รายงานผลโครงการวิจัย

พัฒนาโปรแกรมการคำนวณคะแนน T ปกติ เพื่อกำหนดระดับคะแนนผ่านเครือข่ายอินเตอร์เน็ต

Development of Online Normalized T Score Calculating Software for Grade Assessment.

โดย

รองศาสตราจารย์วัชระ รอดสัมฤทธิ์

นางสาวจันทนี อุทธิสินธุ์

นางสาวชนกนันท์ บางเลี้ยง

สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

(ได้รับเงินงบประมาณประจำปี 2551 หมวดเงินอุดหนุน)

บทคัดย่อ

งานวิจัยสิ่งประดิษฐ์นี้เป็นการสร้างโปรแกรม “TScoreCal” สำหรับใช้ในการคำนวณคะแนนทีปกติและกำหนดระดับคะแนนของนักศึกษา โปรแกรมสามารถทำงานผ่านเครือข่ายอินเทอร์เน็ต หรือทำงานในเครื่องคอมพิวเตอร์ส่วนบุคคล สามารถรองรับจำนวนนักศึกษาได้ 2000 คน

ผู้ใช้งานเพียงแค่นำคะแนนดิบป้อนเข้าไปในโปรแกรม โปรแกรมจะตรวจสอบข้อมูล จัดเรียงลำดับคะแนน คำนวณคะแนนทีปกติ และกำหนดระดับคะแนนในช่วงต่าง ๆ ตามที่ผู้ใช้งานต้องการ

Abstract

This project presents the utility program “TScoreCal” which calculates normalized T score and grade level. This program can runs via internet or in any desktop PC computer and still works efficiently at 2,000 data.

Users only input data by typing (or copying and pasting) into text area box. This program will verifies, sorts data, calculates normalized T Score and sets grade level in any specific range.

สารบัญ

บทที่			หน้า
1	บทนำ		
	1.1	ที่มาและความสำคัญของปัญหา	1
	1.2	ความมุ่งหมายของการวิจัย	1
	1.3	ความสำคัญของการวิจัย	2
	1.4	ขอบเขตของการวิจัย	2
	1.5	นิยามศัพท์เฉพาะ	3
2	เอกสารและงานวิจัยที่เกี่ยวข้อง		5
	2.1	ความรู้เบื้องต้นเกี่ยวกับการวัดผลและประเมินผลการศึกษา	5
	2.2	สถิติเบื้องต้นสำหรับการวัดผลและประเมินผล	10
	2.3	คะแนนและการตัดเกรด	14
3	วิธีการดำเนินการวิจัย		23
4	ผลการวิจัย		27
	4.1	ส่วนที่เป็น input ของข้อมูล	27
	4.2	ส่วนที่คำนวณค่าสถิติพื้นฐาน	32
	4.3	ส่วนที่คำนวณคะแนนทีปกติ และเลือกพิสัยของระดับคะแนน	41
	4.4	ส่วนที่แสดงผลลัพธ์	49
	4.5	ส่วนที่แสดงข้อความแนะนำการใช้งานโปรแกรม	59
	4.6	การนำโปรแกรมไปติดตั้งที่แม่ข่ายอินเตอร์เน็ต	64
	4.7	ผลการทำงานของโปรแกรมผ่านเครือข่ายอินเตอร์เน็ต	69
5	สรุปวิจารณ์และข้อเสนอแนะ		75
บรรณานุกรม			85
ภาคผนวก A	ผลการคำนวณหาพื้นที่ใต้เส้นโค้งปกติ และค่า Z เปรียบเทียบกับผลลัพธ์จากการใช้ฟังก์ชันที่มีอยู่ใน Excel		87

บทที่ หน้า

ภาคผนวก B การคำนวณคะแนนทีปกติ และตัดเกรดโดยใช้ โปรแกรมสำเร็จรูป Excel 99
ภาคผนวก C การสร้างลายเซ็นให้ TScoreCal.jar เพื่อใช้กับ Java Web Start 117
ภาคผนวก D คู่มือการใช้งานโปรแกรม TScoreCal 127
ภาคผนวก E การทดสอบโปรแกรม TScoreCal 137

ดรรชนี 157

สามารถทดสอบหรือใช้งานโปรแกรม ได้ที่

http://203.158.100.140/Tscore/

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

การกำหนดระดับคะแนนในการวัดผลการศึกษาแบบอิงกลุ่ม นิยมใช้วิธีแปลงคะแนนดิบให้เป็นคะแนนทีปกติ (Normalized T Score) ซึ่งต้องนำคะแนนมาเรียงลำดับ หาความถี่ของคะแนนแต่ละค่า หาความถี่สะสม และหาเปอร์เซนต์ไทล์ของคะแนน จากนั้นหาคะแนนทีจากการเปิดตาราง แล้วนำมากำหนดระดับคะแนนในขั้นตอนสุดท้าย ถ้าจำนวนนักศึกษาไม่มากนัก(ไม่เกิน 100 คน) วิธีการดังกล่าวสามารถทำได้ด้วยมือและเครื่องคำนวณ แต่ถ้านักศึกษามีเป็นจำนวนมาก เช่น สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์และเทคโนโลยี ของมหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี จัดการเรียนการสอนวิชาฟิสิกส์ 1 ภาค 1/2549 มีนักศึกษาที่ลงทะเบียนเรียนจำนวน 1,583 คน ภาค 1/2550 มีนักศึกษาลงทะเบียนเรียน 1,528 คน ถ้ายังคงใช้วิธีการดังกล่าว ต้องใช้เวลานานและสิ้นเปลืองแรงงานมิใช่น้อย เมื่อมีจำนวนนักศึกษามาก ๆ โอกาสผิดพลาดยิ่งมากตามไปด้วย ตารางหาคะแนนทีปกติที่มีใช้กันทั่วไปไม่สามารถรองรับจำนวนนักศึกษาที่เกิน 1,000 คนขึ้นไปได้ เพื่อที่จะแก้ปัญหาตรงจุดนี้ จึงสมควรใช้โปรแกรมคอมพิวเตอร์ช่วยในการคำนวณหาคะแนนทีปกติกำหนดระดับคะแนน เพื่อให้ผู้สอนสามารถกำหนดระดับคะแนนได้ทั้งที่บ้านและที่ทำงาน โปรแกรมนี้จึงควรทำงานในลักษณะ Online ผ่านเครือข่ายอินเตอร์เน็ต หรือทำงานได้ในเครื่องคอมพิวเตอร์ส่วนบุคคล (ในกรณีที่ไม่สามารถเชื่อมต่ออินเตอร์เน็ต) โดยติดตั้งโปรแกรมนี้ไว้ที่เครื่องแม่ข่ายที่ทำหน้าที่เป็น Web Server ใด ๆ

1.2 ความมุ่งหมายของการวิจัย

1. เพื่อออกแบบและพัฒนาโปรแกรมคำนวณ Normalized T score. (เรียกชื่อสั้น ๆ ว่า TScoreCal) ผู้ใช้สามารถกำหนดระดับคะแนนโดยใช้งานผ่านเครือข่ายอินเตอร์เน็ตหรือ download โปรแกรมทำงานในลักษณะ application ในเครื่องคอมพิวเตอร์ของผู้ใช้งาน

1.3 ความสำคัญของการวิจัย

1. ได้โปรแกรมคำนวณคำนวณคะแนนทีปกติ (Normalized T Score) เพื่อกำหนดระดับคะแนนในการวัดผล-ประเมินผลนักศึกษา โดยที่ผู้สอนไม่ต้องคำนวณด้วยมือหรือเปิดตารางหาค่าคะแนนทีอีกต่อไป

2. โปรแกรมนี้สร้างขึ้นโดยใช้ภาษาคอมพิวเตอร์ Java ดังนั้น จึงสามารถทำงานผ่านเครือข่ายอินเตอร์เน็ต โดยการแสดงผลผ่าน Browser หรือผู้ใช้สามารถดาวน์โหลดมาไว้ในเครื่องคอมพิวเตอร์ส่วนบุคคลแล้วให้ทำงานในลักษณะ application ได้เช่นกัน

1.4 ขอบเขตของการวิจัย

ในการพัฒนา ครั้งนี้ได้กำหนดขอบเขตความสามารถโปรแกรมไว้ดังนี้

1. กำหนดจำนวนคะแนนนักศึกษาไว้ไม่เกิน 2,000 คน
2. ใช้ภาษาคอมพิวเตอร์ที่เรียกว่า จาวา(Java) ในการเขียนโปรแกรมทั้งหมด
3. สามารถทำงานผ่านเครือข่ายอินเตอร์เน็ตในลักษณะแอพเพล็ต (applet) ซึ่งจะทำงานอยู่ในบราวเซอร์ เพราะข้อจำกัดในเรื่องความปลอดภัยของการทำงานของแอพเพล็ตจึงไม่อนุญาตให้แอพเพล็ตอ่าน และเขียน หรือดูข้อความใด ๆ ของเครื่องที่เป็นลูกข่ายได้ โปรแกรมจึงไม่สามารถอ่านข้อมูลจากไฟล์ที่เก็บไว้ในไดรว์ใด ๆ และไม่สามารถส่งผลลัพธ์ที่ได้พิมพ์ออกทางเครื่องพิมพ์ แต่ผู้ใช้สามารถ copy ผลลัพธ์ที่ได้ แล้วนำมาบันทึกไว้เป็น text file เพื่อใช้ประกอบการพิจารณากำหนดระดับคะแนนภายหลังได้
4. ในกรณีที่ผู้ใช้ต้องการดาวน์โหลดโปรแกรมมาทำงานในเครื่อง ผู้ใช้เพียงแต่คลิกที่ลิงค์(link) ที่ระบุไว้ ในการพัฒนานี้ได้อาศัย Java Network Launching Protocol (JNLP file) ซึ่งจะช่วยในการติดตั้งให้ดำเนินไปอย่างอัตโนมัติโดยผู้ใช้งานคลิกเมาส์เพียงครั้งเดียว การทำงานของโปรแกรมในกรณีนี้จะทำงานนอกบราวเซอร์ สามารถอ่านข้อมูลจากไฟล์ (ในกรณีที่มีนักศึกษาเป็นจำนวนมาก และพิมพ์คะแนนเก็บเป็นไฟล์ไว้) และสามารถส่งผลลัพธ์ที่ได้จากการคำนวณไปพิมพ์ที่เครื่องพิมพ์

1.5 นิยามศัพท์เฉพาะ

คะแนนทีปกติ (Normalized T Score) หมายถึงคะแนนทีที่ได้จากการแปลงคะแนนดิบให้สามารถเปรียบเทียบตำแหน่งของคะแนนโดยเทียบกับพื้นที่ใต้เส้นโค้งปกติ พื้นที่ใต้โค้งปกติตรงตำแหน่งที่คะแนนนั้นอยู่ จากสูตร

$$\text{พื้นที่ใต้โค้ง} = \frac{\text{ความถี่สะสมของคะแนนชั้นที่อยู่ต่ำกว่า 1 ชั้น + 0.5 ของความถี่คะแนนในชั้นนั้น}}{\text{จำนวนผู้เรียนทั้งหมด}}$$

พื้นที่ใต้โค้งจะบอกตำแหน่งของผู้ที่ได้คะแนนนั้นว่าอยู่เหนือกว่าผู้อื่นอยู่เท่าใด เป็นการบอกเปอร์เซ็นต์ไทล์ (Percentile) นั่นเอง

การแปลงคะแนนเช่นนี้จะทำให้ลดความคลาดเคลื่อนอันเนื่องมาจากการกระจายของคะแนนไม่เป็นเส้นโค้งปกติ ซึ่งอาจเกิดจากข้อสอบไม่ได้มาตรฐาน ทำให้เส้นโค้งแสดงการกระจายของคะแนนอาจเบ้ซ้ายหรือเบ้ขวาก็ได้

คะแนนที (ไม่มีคำว่าปกติ) จะหาได้จากการหาคะแนน z ก่อน โดยที่ z

$$z = \frac{\text{คะแนนที่ได้} - \text{คะแนนเฉลี่ยของกลุ่ม}}{\text{ส่วนเบี่ยงเบนมาตรฐานของคะแนน}}$$

จากนั้นจึงนำค่า z มาหาคะแนนทีอีกทอดหนึ่งจากสูตร

$$T = \text{คะแนน } z \times 10 + 50$$

การหาคะแนนที โดยวิธีการเช่นนี้ ความเบ้ของเส้นโค้งการกระจายของคะแนนเป็นอย่างไรก็จะถูกถ่ายทอดมายังคะแนนทีโดยตรง ผลก็คือระดับคะแนนหรือเกรดที่ได้จะไม่สอดคล้องกับกรณีที่เป็นเส้นโค้งปกติ อาจส่งผลให้มีผู้ได้ระดับคะแนน A หรือ F หรือระดับคะแนนใดคะแนนหนึ่งมากเกินไป

จาวา (Java) เป็นทั้งภาษาคอมพิวเตอร์และสภาพแวดล้อมที่ทำให้โปรแกรมที่ถูกคอมไพล์แล้วทำงานได้โดยลำพัง (Run-Time Environment, JRE) ออกแบบและสร้างขึ้นโดยบริษัทซัน ไมโครซิสเท็ม(Sun Micro System) เมื่อปี ค.ศ. 1996 ปัจจุบัน (เดือนมิถุนยน 2551) พัฒนามาถึง version 6 ภาษาจาวาจัดเป็นภาษาเชิงวัตถุ (Object-oriented language) คล้ายกับภาษา C++ แต่ออกแบบให้มีความง่าย และลดช่องทางที่จะทำให้เกิดความผิดพลาดได้ดีกว่า คอมไพเลอร์หรือตัวแปลภาษาจาวาจะแปลงชุดคำสั่งให้อยู่ในรูป byte codes ไม่ว่าคอมพิวเตอร์นั้นจะเป็น CPU ชนิดใด หรือทำงานภายใต้ระบบปฏิบัติการใด(Windows, Linux, Solaris,

MacOS) ขอเพียงแต่ให้คอมพิวเตอร์เครื่องนั้นมีการติดตั้ง Java Virtual Machine (JVM) อยู่ภายในเครื่องนั้น ก็สามารถทำให้ byte codes นั้นทำงานได้ทันที โดยไม่ต้องติดตั้งคอมไพเลอร์ให้ยุ่งยาก คุณลักษณะของจาวาเช่นนี้ทำให้การเขียนภาษาจาวา สามารถนำโปรแกรมที่ไปใช้งานได้ทุก platform โดยไม่ต้องคอมไพล์ใหม่หรือแก้ไขโปรแกรมเลย

แอพเพล็ต (Applet) เป็นโปรแกรมขนาดเล็กที่สามารถทำงานได้ในบราวเซอร์ โดยที่บราวเซอร์นั้นจะต้องมี JRE (Java Runtime Environment) ติดตั้งเป็น plug in แอพเพล็ตจะถูกเขียนด้วยภาษาจาวา แล้วคอมไพล์เป็น class file เมื่อแอพเพล็ตถูกเรียกใช้งาน บราวเซอร์จะดาวน์โหลดแอพเพล็ตจากแม่ข่ายมาไว้ที่เครื่องลูกข่าย และทำงานโดยอาศัยทรัพยากรของเครื่องลูกข่าย เป็นการลดภาระงานการประมวลผลของเครื่องแม่ข่ายในกรณีที่มีผู้เข้าถึงหรือมีผู้เข้ามาใช้ข้อมูลในเครื่องแม่ข่ายเป็นจำนวนมาก ๆ

ข้อจำกัดของการทำงานของแอพเพล็ตคือไม่สามารถเขียนโปรแกรมให้แอพเพล็ตอ่าน เขียน เปลี่ยนแปลง ลบ หรือดูข้อมูลของเครื่องลูกข่ายได้ เพราะป้องกันผู้มีเจตนาร้ายที่จะใช้แอพเพล็ตล้วงความลับหรือทำลายข้อมูลในเครื่องคอมพิวเตอร์ของลูกข่าย ไม่สามารถไปดึงโปรแกรมใดที่อยู่ภายในเครื่องลูกข่ายให้ทำงานได้ อีกทั้งไม่สามารถสามารถติดต่อไปยังระบบเครือข่ายอื่น ๆ ได้ ยกเว้น เครื่องแม่ข่ายที่เก็บแอพเพล็ตนั้นเท่านั้น

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการจัดการเรียนการสอน มีองค์ประกอบที่สำคัญ 3 ส่วน ได้แก่ ผู้สอน กระบวนการหรือวิธีการสอน และผู้เรียน อาจเปรียบเทียบว่าตัวผู้เรียนเป็นตัวป้อน (Input) วิธีการสอนและตัวผู้สอน (process) เป็นกระบวนการ และผลสัมฤทธิ์ทางการเรียนของผู้เรียนเป็นผลผลิต (Output) นั่นเอง

ผู้เรียนจึงเป็นผู้ได้รับผลกระทบโดยตรง การตัดสิน หรือประเมินค่าผลผลิต (ผู้เรียน)เป็นเรื่องสำคัญและละเอียดอ่อน ซึ่งนั่นหมายถึงการตัดสินพิพากษาชีวิตและอนาคตของผู้เรียน ดังนั้นผู้สอน จะต้องมีความรับผิดชอบและมีมโนธรรมและคำนึงถึงสิ่งที่จะเกิดขึ้นตามมาด้วย

ในที่นี้จะสรุปสาระเกี่ยวกับการวัดผลและการประเมินผลการเรียนการสอนอย่างพอสังเขป เพื่อเป็นแนวทางในการที่จะพิจารณาตัดสินผลการเรียน เริ่มตั้งแต่ความรู้เบื้องต้นเกี่ยวกับการวัดผลและการประเมินผล สถิติเบื้องต้นที่ใช้สำหรับการวัดผลและประเมินผล ตลอดจนสาระเกี่ยวกับคะแนนและการกำหนดระดับคะแนน

2.1 ความรู้เบื้องต้นเกี่ยวกับการวัดผลและประเมินผลการศึกษา

2.1.1 ความหมายของการวัดผล

การวัดผล (Measurement) หมายถึงกระบวนการหรือวิธีการเพื่อกำหนดตัวเลขหรือจำนวนต่าง ๆ หรือสัญลักษณ์ ให้กับบุคคล สิ่งของ หรือเหตุการณ์ต่าง ๆ โดยใช้เครื่องมืออย่างใดอย่างหนึ่งวัด เพื่อให้ได้ข้อมูลที่แทนปริมาณ หรือขนาดหรือคุณสมบัติของสิ่งที่ต้องการวัด การวัดผลมีองค์ประกอบที่สำคัญ 3 ประการคือ

1) สิ่งที่จะวัดหรือปัญหา เช่น
   - เด็กชายคาวีสอบได้กี่คะแนน
   - คุณแม่ของนารินมีอาชีพอะไร
   - หมายเลขโทรศัพท์ของเติมศักดิ์หมายเลขอะไร
2) เครื่องมือวัด หรือเทคนิควิธีการในการรวบรวมข้อมูล เช่น
   - แบบสอบถาม
   - แบบทดสอบ
   - การสัมภาษณ์
3) ผลการวัด ข้อมูลเชิงปริมาณหรือเชิงคุณภาพ เช่น
   - 70 คะแนน

- อาชีพรับราชการ

- 02-549-3891

2.1.2 ความหมายของการประเมินผล

การประเมินผล (Evaluation) หมายถึงการตัดสินคุณค่าหรือวินิจฉัยสิ่งต่าง ๆ ที่ได้จากการวัดผล โดยเปรียบเทียบกับผลการวัดหรืออาศัยเกณฑ์การพิจารณาอย่างใดอย่างหนึ่ง องค์ประกอบของการประเมินผล ประกอบด้วย ข้อมูล เกณฑ์ และการตัดสินคุณค่า

1) ข้อมูล เช่น

- คาวีสอบได้ 70 คะแนน
- วิชัย สอบได้ 40 คะแนน
- ชัยวัฒน์ สอบได้ 80 คะแนน
- วัฒนา สอบได้ 55 คะแนน

2) เกณฑ์ เช่นเปรียบเทียบกับเกณฑ์ที่กำหนดคือ

- คะแนนมากกว่า 80 ได้เกรด A
- คะแนนระหว่าง 70-79 ได้เกรด B
- คะแนนระหว่าง 60-69 ได้เกรด C
- คะแนนระหว่าง 50-59 ได้เกรด D
- คะแนนต่ำกว่า 50 ได้เกรด F

3) การตัดสินคุณค่าหรือการตัดสินใจ

- คาวีได้เกรด B
- วิชัย ได้เกรด F
- ชัยวัฒน์ ได้เกรด A
- วัฒนา ได้ เกรด D

จากความหมายของการวัดผลและการประเมินผล สรุปความสัมพันธ์ได้ดังนี้

การประเมินผล = การวัดผล + การตัดสินคุณค่า

2.1.3 ธรรมชาติของการวัดผลการศึกษา

การวัดผลการศึกษามีขีดจำกัด หลายประการดังนี้

1) การวัดผลการศึกษาเป็นการวัดที่ไม่สมบูรณ์(Incomplete) ไม่สามารถวัดได้ละเอียดครบถ้วนตามที่ต้องการ เพราะเป็นการวัดในสิ่งที่เป็นนามธรรม ไม่สามารถทราบปริมาณหรือขอบเขตที่แน่นอน เช่นรายวิชาที่สอนนั้น มีรายละเอียดมากมายจนเกินความสามารถที่จะวัดได้

หมด ผู้สอนจึงจำเป็นต้องเลือกเนื้อหาสาระเพียงบางส่วน เป็นตัวแทนในรายวิชานั้น ๆ มาทำการวัด ผลที่ได้ก็จะคลาดเคลื่อนได้ง่าย

2) การวัดผลการศึกษาเป็นการวัดทางอ้อม (Indirect) เพราะไม่มีเครื่องมือใด ๆ ที่จะวัดพฤติกรรมของนักเรียนที่เปลี่ยนแปลงไป ภายหลังจากการได้รับการศึกษาซึ่งเป็นสิ่งที่มองไม่เห็นจับต้องไม่ได้ จึงไม่สามารถวัดได้โดยตรง ต้องหาวิธีการต่าง ๆ มาวัดแล้วแปลความหมายของพฤติกรรมนั้น ๆ ออกมาอีกทอดหนึ่ง

3) การวัดผลการศึกษามีความคลาดเคลื่อน(Error) เพราะพฤติกรรมที่ทำการวัดส่วนใหญ่มีลักษณะซับซ้อนสังเกตหรือจับต้องไม่ได้ เช่นความรู้ ความเชื่อ ความคิดเห็น

นอกจากนี้ยังมีปัจจัยที่ส่งผลให้เกิดความคลาดเคลื่อนในการวัด ได้แก่ ตัวผู้ถูกวัดหรือตัวผู้เรียนอาจสุขภาพไม่ดี มีความกังวล ทุจริตการสอบ ฯลฯ และปัจจัยทางด้านสิ่งแวดล้อม เช่น สภาพห้องไม่ดี มีเสียงรบกวน ข้อสอบบกพร่อง

4) ผลจากการวัดแสดงออกในรูปของความสัมพันธ์(Relation) ในการวัดแล้วได้ข้อมูลเพียงอย่างเดียวย่อมไม่มีความหมายใด ๆ เช่นนายนาวินได้คะแนนสอบ 40 คะแนน ไม่สามารถบอกได้ว่า ได้คะแนนมาก-น้อย หรือเก่ง-อ่อน หากจะให้คะแนนตัวนี้มีความหมายหรือเกิดความเข้าใจได้ต้องนำไปเปรียบเทียบกับข้อมูลอย่างอื่น เช่นนำไปเปรียบเทียบกับคะแนนเต็ม หรือนำไปเปรียบเทียบกับคะแนนของกลุ่ม หรือนำไปเปรียบเทียบกับเกณฑ์ที่กำหนดไว้

5) การวัดผลการศึกษาเป็นการวัดที่ไม่มีศูนย์แท้ หรือศูนย์สัมบูรณ์(Absolute zero) กล่าวคือผู้ที่สอบได้ 0 คะแนน ไม่ได้หมายความว่าไม่มีความรู้ในเรื่องนั้น เขาอาจมีความรู้แต่ข้อสอบไม่ได้ถามสิ่งที่เขารู้

2.1.4 การประเมินผลแบบอิงเกณฑ์และอิงกลุ่ม

การวัดผลและประเมินผลการศึกษาเป็นศาสตร์ขั้นสูง เพราะมีการจัดระบบเนื้อหาความรู้เป็นขั้นตอน มีระเบียบกฎเกณฑ์ และแบบแผนวิธีการ สามารถพิสูจน์ทดลองความจริงและตรวจสอบผลได้

การวัดผลและการประเมินผลเป็นการทดสอบเพื่อค้นและพัฒนาศักยภาพของมนุษย์ การวัดผลประเมินผลผู้เรียนก็เพื่อค้นหาสมรรถภาพของผู้เรียนว่า ใครมีอะไร ใครไม่มีอะไร และมากน้อยแค่ไหน เพื่อพัฒนาสมรรถภาพของผู้เรียน โดยการส่งเสริมสิ่งที่มีอยู่ให้มากขึ้นและถ้าไม่มีก็ปลูกฝังสิ่งใหม่ขึ้นมา

การวัดผลและการประเมินผลจำแนกตามวัตถุประสงค์ของการประเมินแบ่งออกเป็น 3 ประเภท

1) การประเมินผลก่อนเรียน (Pre-evaluation) เพื่อตรวจสอบความรู้พื้นฐานและทักษะของผู้เรียน ว่ามีความรู้เพียงพอที่จะเรียนต่อในรายวิชาใหม่ หรือเนื้อหาใหม่ได้หรือไม่

2) การประเมินผลระหว่างเรียน (Formative evaluation) เพื่อตรวจสอบว่าผู้เรียน บรรลุตามวัตถุประสงค์การเรียนที่กำหนดไว้หรือไม่เพียงใด

3) การประเมินผลรวมสรุป (Summative evaluation) เพื่อตัดสินผลการเรียน มีจุดมุ่งหมาย เพื่อศึกษาว่าผู้เรียนมีความรู้มากน้อยเพียงใด ควรตัดสินได้-ตก ผ่าน-ไม่ผ่าน หรือควรได้เกรดอะไร

นอกจากนี้ยังจำแนกประเภทของการประเมินผล ตามระบบการวัดผล แบ่งได้ 2 ประเภท คือการประเมินผลแบบอิงเกณฑ์และการประเมินผลแบบอิงกลุ่ม

การประเมินผลแบบอิงเกณฑ์ (Criterion reference)

การประเมินผลแบบอิงเกณฑ์หมายถึงการประเมินที่มุ่งนำเอาผลการเรียนของผู้เรียนแต่ละคนมาเปรียบเทียบกับเกณฑ์ ที่กำหนดขึ้น โดยไม่ต้องเปรียบเทียบกับผู้เรียนคนอื่น ๆ เป็นการประเมินที่ต้องการทราบสถานภาพของบุคคลโดยอาศัยเกณฑ์ที่กำหนดไว้ในจุดมุ่งหมายเป็นหลัก

การประเมินแบบอิงเกณฑ์ มีจุดมุ่งหมายเพื่อบ่งชี้สถานภาพของผู้เรียนแต่ละคน เพื่อเปรียบเทียบกับเกณฑ์ ที่ทดสอบเพื่อตัดสินว่าผู้เรียนบรรลุตามวัตถุประสงค์ที่กำหนดไว้หรือไม่และมากน้อยเพียงใด อันจะนำไปสู่การปรับปรุงการเรียนการสอนเมื่อผู้เรียนไม่สามารถทำข้อสอบได้ถึงเกณฑ์ ต้องมีการสอนซ่อมเสริมจนกว่าจะผ่านเกณฑ์ การประเมินผลแบบอิงเกณฑ์จึงเหมาะสำหรับการเรียนการสอนในห้องเรียน

การประเมินผลแบบอิงกลุ่ม(Norm reference)

การประเมินผลแบบอิงกลุ่ม หมายถึงการประเมินที่มุ่งนำผลการประเมินมาจำแนกผู้เรียนออกตามความสามารถโดยพิจารณาจากการเปรียบเทียบผลการประเมินของผู้เรียนแต่ละคน กับกลุ่มผู้เรียนด้วยกัน โดยมีแนวคิดเกี่ยวกับความแตกต่างระหว่างบุคคล เพื่อจะทราบว่าผู้เรียนแต่ละคนมีความสามารถ มากหรือน้อยกว่าคนอื่น ๆ ในกลุ่มเดียวกัน

การประเมินผลแบบอิงเกณฑ์และแบบอิงกลุ่มมีข้อแตกต่างกันและมีข้อดีข้อเสียดังนี้

ความแตกต่างระหว่างการประเมินผลแบบอิงเกณฑ์และแบบอิงกลุ่ม

แบบอิงเกณฑ์	**แบบอิงกลุ่ม**
1. ยึดหลักการเรียนเพื่อรอบรู้ กล่าวคือผู้เรียนทุกคนสามารถเรียนรู้จนบรรลุเป้าหมาย	1. ยึดหลักการความแตกต่างระหว่างบุคคล กล่าวคือผู้เรียนทุกคนเรียนรู้ได้ไม่เท่ากัน
2. การวัดความสามารถของผู้เรียนแต่ละคนโดยเปรียบเทียบกับเกณฑ์หรือมาตรฐาน	2. การวัดความสามารถของผู้เรียนแต่ละคนมีจุดมุ่งหมายเพื่อเปรียบเทียบกับเพื่อนในกลุ่มเดียวกัน
3. การสร้างแบบทดสอบ เขียนตามจุดประสงค์ทั่วไป และต้องสุ่มเนื้อหามาอย่างดีและใช้แบบทดสอบชุดเดียวกันทุกคน	3. การสร้างแบบทดสอบเขียนตามวัตถุประสงค์เชิงพฤติกรรม ไม่จำเป็นต้องใช้แบบทดสอบชุดเดียวกับทุกคน
4. คุณภาพของแบบทดสอบเน้นความเที่ยงตรงเชิงเนื้อหาและโครงสร้าง ไม่คำนึงถึงความยากง่ายของข้อสอบ เน้นการวิเคราะห์ผลจากการวัดก่อนเรียนและหลังเรียนว่าผู้เรียนมีความรอบรู้แล้ว กับไม่รอบรู้แตกต่างกันอย่างไร	4. คุณภาพของแบบทดสอบเน้นความยากง่ายและอำนาจจำแนกของข้อสอบ วิเคราะห์ข้อสอบรายข้อใช้เกณฑ์ภายในคือกลุ่มผู้ได้คะแนนสูงกับกลุ่มผู้ได้คะแนนต่ำ
5. การแปลความหมายของคะแนนโดยใช้คะแนนดิบ ลักษณะและรูปแบบของคะแนนไม่มีความสำคัญ บอกได้เพียงว่าผู้เรียนมีความสามารถเท่าใด เมื่อเทียบกับเกณฑ์	5. การแปลความหมายของคะแนนโดยใช้คะแนนมาตรฐาน ลักษณะและรูปแบบการกระจายของคะแนน เป็นสาระที่จะต้องพิจารณา ควรมีการกระจายมากเพื่อให้ได้ค่าอำนาจจำแนกที่สูง บอกได้ว่าผู้เรียนมีความสามารถเท่าใด เมื่อเทียบกับปริมาณทั้งหมดที่ทดสอบ

ข้อดีและข้อเสียของการประเมินผลแบบอิงเกณฑ์

ข้อดี	ข้อเสีย
1. เป็นการประเมินเพื่อปรับปรุงการเรียนการสอน	1. วิธีการสอบยุ่งยาก
2. ส่งเสริมให้มีการช่วยเหลือกันมากกว่าการแข่งขัน	2. ไม่เน้นความยากง่ายของข้อสอบ อาจทำให้คุณภาพการเรียนการสอนต่ำได้
3. เป็นการประเมินทั้งผู้เรียนและกระบวนการเรียนการสอน	3. การกำหนดเกณฑ์ที่เหมาะสมทำได้ยาก

ข้อดีและข้อเสียของการประเมินผลแบบอิงกลุ่ม

ข้อดี	ข้อเสีย
1. ส่งเสริมให้เกิดพัฒนาตนเอง	1. เกิดการแข่งขันกันมากกว่าการช่วยเหลือกัน
2. สะดวกในการออกข้อสอบ เพราะข้อสอบชุดเดียวกัน	2. ไม่สามารถนำคะแนนไปใช้อิงเกณฑ์ได้
3. ใช้ได้ดีในการสอบคัดเลือกและการพยากรณ์	3. เป็นการประเมินผลผู้เรียนเป็นส่วนใหญ่ ไม่เน้นกระบวนการเรียนการสอน

2.2. สถิติเบื้องต้นสำหรับการวัดผลและประเมินผล

ในการวัดผลและประเมินผลใด ๆ ก็ตาม การเลือกใช้สถิติที่ถูกต้องสำหรับการวิเคราะห์ผลที่ได้จากการวัด เป็นสิ่งสำคัญ เพื่อจะทำให้การประเมินผลมีความถูกต้องเชื่อถือได้ และสามารถนำผลการประเมินไปใช้ประโยชน์ได้

การนำข้อมูลมาวิเคราะห์ทางสถิติต้องเป็นข้อมูลดิบ(Raw score) ซึ่งเป็นข้อมูลเชิงปริมาณที่รวบรวมได้มาแล้วแต่ยังไม่ได้นำมาจัดระเบียบ

สถิติที่ใช้ในการวิเคราะห์ข้อมูลสำหรับการวัดผลและประเมินผลได้แก่ การแจกแจงความถี่ การวัดแนวโน้มเข้าสู่ศูนย์กลาง การวัดการกระจาย และสหสัมพันธ์

2.2.1 การแจกแจงความถี่ (Frequency distribution)

ข้อมูลที่ได้จากการทดสอบ เป็นข้อมูลดิบที่ยังไม่ได้จัดระบบ มักจะมีตัวเลขที่มีค่าสูงและมีค่าต่ำปะปนกันอย่างไม่เป็นระบบ ทำให้การพิจารณาหรือคำนวณค่าสถิติไม่สะดวก ต้องมีการนำ

ข้อมูลหรือคะแนนที่เกิดจากการสอบ มาจัดเรียงใหม่ จากคะแนนสูงไปต่ำ หรือจากคะแนนต่ำไปสูง เรียกว่าการแจกแจงความถี่ เพื่อช่วยให้สะดวกในการพิจารณา การแจกแจงความถี่แบ่งออกเป็น 2 แบบ

1) การแจกแจงความถี่แบบไม่จัดหมวดหมู่ข้อมูล (Ungrouped data) ได้แก่การเรียงคะแนนจากสูงไปต่ำ และรอยขีดคะแนน (tally)

2) การแจกแจงแบบจัดหมวดหมู่ข้อมูล (Grouped data) กรณีที่มีข้อมูลจำนวนมากและคะแนนสูงสุดกับคะแนนต่ำสุดห่างกันมาก จึงต้องแจกแจงโดยจัดเป็นช่วงคะแนน มีขั้นตอนการทำดังนี้

- หาพิสัย(Range) คือความแตกต่างระหว่างคะแนนสูงกับคะแนนต่ำ

พิสัย = คะแนนสูง – คะแนนต่ำ

- กำหนดชั้นคะแนนที่ต้องการ
- หาช่วงความกว้างของแต่ละชั้น หรืออันตรภาคชั้น(Interval) จากสูตร

$$I = \frac{R}{N}$$

เมื่อ I = อันตรภาคชั้น

R = พิสัย

N = จำนวนชั้น

- เขียนขีดจำกัดของคะแนนโดยขึ้นต้นด้วยชั้นของคะแนนสูงไปจนถึงชั้นของคะแนนต่ำ
- ขีดรอยคะแนนแล้วนับจำนวนรอยขีดใส่ในช่วงความถี่

2.2.2 การวัดแนวโน้มเข้าสู่ศูนย์กลาง (Measures of control tendency)

การวัดแนวโน้มเข้าสู่ศูนย์กลางเป็นการหาค่ากลาง เพื่อเป็นตัวแทนของข้อมูล ช่วยให้ทราบลักษณะของข้อมูลทั้งชุดว่าเป็นอย่างไร ค่าสถิติที่ใช้ในการหาแนวโน้มเข้าสู่ศูนย์กลาง ได้แก่ ค่าเฉลี่ย มัธยฐาน ฐานนิยม

1) **ค่าเฉลี่ย(Mean)** หรือตัวกลางเลขคณิต หมายถึงค่าที่ได้จากการนำเอาข้อมูลทั้งหมดมารวมกัน แล้วหารด้วยจำนวนข้อมูลทั้งหมด จะได้ว่า

$$\overline{X} = \frac{\sum X}{N}$$

เมื่อ $\overline{X}$ = ค่าเฉลี่ย

$\sum X$ = ผลรวมของคะแนนทั้งหมด

N = จำนวนข้อมูลทั้งหมด

ค่าเฉลี่ยเป็นค่าที่ใช้กันมากที่สุด ซึ่งค่าเฉลี่ยมีคุณสมบัติดังนี้

- ผลรวมของค่าเบี่ยงเบนของข้อมูลแต่ละตัว จากค่าเฉลี่ยจะมีค่าเป็นศูนย์
- ถ้านำค่าคงที่มาบวก ลบ คูณ หาร กับคะแนนทุกตัวของข้อมูลชุดหนึ่ง ด้วยค่าคงที่นั้น

2) **มัธยฐาน** (Median) มัธยฐานหมายถึงข้อมูลที่อยู่ในตำแหน่งกึ่งกลางของข้อมูลทั้งชุด ซึ่งเรียงค่าข้อมูลแล้ว ซึ่งมัธยฐานเป็นวิธีการหาค่ากลางของข้อมูล ที่เบ้ไปทางใดทางหนึ่ง หรือข้อมูลที่ทราบแต่ช่วงกลาง ๆ ไม่ทราบค่าสูงและค่าต่ำ

3) **ฐานนิยม** (Mode) ฐานนิยมหมายถึงข้อมูลที่มีค่าซ้ำกันมากที่สุด หรือข้อมูลที่มีความถี่สูงสุดในข้อมูลชุดหนึ่ง ๆ ซึ่งฐานนิยมจะใช้กับข้อมูลที่มีการแจกแจงความถี่ และไม่มีการแจกแจงความถี่

การเปรียบเทียบค่าเฉลี่ย มัธยฐาน และฐานนิยม

- ข้อมูลชุดเดียวกันที่มีค่าเฉลี่ย ค่ามัธยฐานและค่าฐานนิยม เท่ากันจะมีการแจกแจงแบบปกติ
- ข้อมูลที่มีค่าเฉลี่ยน้อยกว่า ค่ามัธยฐานและค่าฐานนิยม จะมีการแจกแจงเบ้ไปทางซ้าย
- ข้อมูลที่มีค่าเฉลี่ยมากกว่า ค่ามัธยฐานและค่าฐานนิยม จะมีการแจกแจงเบ้ไปทางขวา
- กรณีที่มีข้อมูลมาก ๆ ต้องการหาค่ากลางอย่างรวดเร็ว ควรใช้ฐานนิยม แต่ถ้าข้อมูลแตกต่างกันมาก หรือมีการแจกแจงเป็นรูปโค้งเบ้ไปทางใดทางหนึ่ง ควรใช้ค่ามัธยฐาน

2.2.3 **การวัดการกระจาย** (Measures of variability)

การกระจายหมายถึงวิธีการทางสถิติ ที่ใช้ในการหาค่าที่แสดงการกระจายของข้อมูล ซึ่งเป็นวิธีการที่ทำให้ทราบว่าข้อมูลชุดหนึ่งมีความแตกต่างกันมากน้อยเพียงใด

วิธีการวัดการกระจายที่นิยมใช้ ได้แก่ พิสัย ส่วนเบี่ยงเบนเฉลี่ย ส่วนเบี่ยงเบนมาตรฐาน และความแปรปรวน

1) **พิสัย**(Range) การหาพิสัยเป็นการหาการกระจายของข้อมูลอย่างคร่าว ๆ ซึ่งเป็นค่าความแตกต่าง ระหว่างคะแนนสูงสุดกับต่ำสุดของข้อมูลชุดหนึ่ง ๆ นิยมใช้กับการพิจารณาข้อมูลที่มีจำนวนน้อย

2) **ส่วนเบี่ยงเบนเฉลี่ย** (Mean deviation) ส่วนเบี่ยงเบนเฉลี่ย เป็นค่าเฉลี่ยของคะแนนที่เบี่ยงเบนออกจากคะแนนเฉลี่ยของข้อมูลชุดนั้น ซึ่งเป็นการวัดการกระจายที่ละเอียดกว่าการหาพิสัย แต่มีข้อจำกัดคือ ค่าส่วนเบี่ยงเบนเฉลี่ยที่ได้ มีทั้งค่าบวกและค่าลบ ไม่เหมาะที่จะนำไปใช้อธิบายข้อมูล หรือนำไปใช้คำนวณหาค่าสถิติขั้นต่อ ๆ ไป

ส่วนเบี่ยงเบนมาตรฐานและความแปรปรวน(Standard deviation and variance) ส่วนเบี่ยงเบนมาตรฐาน เป็นการวัดการกระจายที่ดี และใช้กันมากที่สุด หาได้จากรากที่สองของค่าเฉลี่ย ของผลรวมของคะแนนทุกค่าที่เบี่ยงเบนไปจากค่าเฉลี่ยยกกำลังสองหาได้จากสูตร

$$S = \sqrt{\frac{\Sigma(X-\overline{X})^2}{N-1}}$$

เมื่อ S = ส่วนเบี่ยงเบนมาตรฐานของกลุ่มตัวอย่าง

X = ข้อมูลหรือคะแนนแต่ละตัว

$\overline{X}$ = คะแนนเฉลี่ยของกลุ่มตัวอย่าง

N = จำนวนข้อมูล

ความแปรปรวนเป็นค่าเฉลี่ยของผลรวมของคะแนนทุกค่า ที่เบี่ยงเบนไปจากค่าเฉลี่ยยกกำลังสอง ความแปรปรวนเป็นการวัดความกระจายในรูปของพื้นที่ หาได้จากสูตร

$$S^2 = \frac{\Sigma(X-\overline{X})^2}{N-1}$$

เมื่อ S^2 = คะแนนแปรปรวน

X = คะแนนแต่ละตัว

$\overline{X}$ = คะแนนเฉลี่ย

N = จำนวนคนทั้งหมด

2.2.4 **สหสัมพันธ์** (Correlation)

สหสัมพันธ์ เป็นสถิติที่ใช้หาความสัมพันธ์ ซึ่งจะเป็นดัชนีบอกทิศทางและขนาดของความสัมพันธ์ระหว่างตัวแปรต่าง ๆ

สหสัมพันธ์มีหลายชนิดขึ้นกับจำนวนตัวแปร และระดับของการวัดตัวแปร ในที่นี้จะกล่าวถึงสหสัมพันธ์อย่างง่าย(Simple correlation) หรือเรียกสหสัมพันธ์แบบเพียร์สัน ซึ่งเป็นสหสัมพันธ์ที่ชี้ความสัมพันธ์ระหว่างตัวแปร 2 ตัว หรือคะแนน 2 ชุด ที่เกิดจากสิ่งเดียวกัน ว่าคล้อยตามกันหรือสอดคล้องกันหรือไม่ เช่นนักศึกษากลุ่มหนึ่งสอบวิชาฟิสิกส์ ได้คะแนน X และสอบวิชาแคลคูลัสได้คะแนน Y หาค่าสหสัมพันธ์อย่างง่าย จากสูตร

$$r_{XY} = \frac{N\Sigma XY - \Sigma X\Sigma Y}{\sqrt{\left[N\Sigma X^2 - (\Sigma X^2)\right]\left[N\Sigma Y^2 - (\Sigma Y^2)\right]}}$$

เมื่อ r_{XY} = ค่าสหสัมพันธ์ระหว่างคะแนนชุด X กับ Y

$\sum X$ = ผลรวมทั้งหมดของคะแนน X

$\sum Y$ = ผลรวมทั้งหมดของคะแนน Y

$\sum X^2$ = ผลรวมทั้งหมดของคะแนน X แต่ละตัวยกกำลังสอง

$\sum Y^2$ = ผลรวมทั้งหมดของคะแนน Y แต่ละตัวยกกำลังสอง

$\sum XY$ = ผลรวมทั้งหมดของคะแนน Xและ Y คูณกันแต่ละคู่

N = จำนวนคนทั้งหมด

การพิจารณาสหสัมพันธ์จากข้อมูล 2 ชุด

- ถ้าค่า r_{XY} เป็นบวกแสดงว่าตัวแปร 2 ชุด มีลักษณะคล้อยตามกัน เช่น นักศึกษาคนใดที่เรียนเก่งวิชาหนึ่ง ก็มีแนวโน้มจะเก่งอีกวิชาหนึ่งด้วย

- ถ้าค่า r_{XY} = 1.00 แสดงว่าตัวแปรทั้ง 2 ชุด คล้อยตามกันอย่างสมบูรณ์ กล่าวคือ นักศึกษาที่เก่งวิชาหนึ่งจะเก่งอีกวิช่หนึ่งด้วย

- ถ้าค่า r_{XY} เป็นลบ แสดงว่าตัวแปร 2 ชุด มีลักษณะตรงข้าม เช่นนักศึกษาเรียนอ่อนวิชาหนึ่งก็มีแนวโน้มจะเก่งอีกวิชาหนึ่ง

- ถ้าค่า r_{XY} เข้าใกล้ศูนย์ แสดงว่าตัวแปร 2 ชุด มีแนวโน้มจะไม่สัมพันธ์กัน มีทิศทางไม่แน่นอน มีแนวโน้มคล้อยตามกันบ้าง และตรงกันข้ามบ้าง

- ถ้าค่า r_{XY} = 0.00 แสดงว่าตัวแปรทั้ง 2 ชุด ไม่สัมพันธ์กันอย่างสมบูรณ์

2.3 **คะแนนและการตัดเกรด**

คะแนนจากการสอบวัดผู้เรียน เป็นตัวที่จะแปลความหมายหรือสะท้อนคุณลักษณะของผู้ที่ได้รับการสอบวัด ซึ่งจะบ่งบอกความสามารถของผู้เรียน ตลอดจนแสดงถึงประสิทธิภาพการสอนของผู้สอนด้วยว่า หลังสอนแล้วผู้เรียนได้พัฒนา หรือเปลี่ยนแปลงพฤติกรรม ไปในลักษณะใด มากน้อยเพียงใด

ดังนั้นผู้ทำหน้าที่วัดผลและประเมินผลจำเป็นจะต้องมีความรู้เกี่ยวกับคะแนน วิธีการแปลความหมายตลอดจนวิธีการให้ระดับคะแนน ที่จะต้องคำนึงถึงความคลาดเคลื่อนที่อาจจะเกิดขึ้นด้วย

2.3.1 **คะแนน** (Score)

คะแนน (Score) คะแนนหมายถึงตัวเลข ที่เกิดจากการวัดพฤติกรรมด้านต่าง ๆ โดยใช้เครื่องมือวัด เป็นตัวเลขที่กำหนดให้กับพฤติกรรมหรือสิ่งที่ถูกสอบวัด กล่าวโดยสรุปคือคะแนนก็คือขนาดของความสามารถในการเรียนรู้ของผู้เรียนนั่นเอง

คะแนนแบ่งได้ 2 ประเภทคือ คะแนนดิบและคะแนนแปลงรูป

คะแนนดิบ (Raw score)

คะแนนดิบเป็นคะแนนที่เกิดจากการสอบโดยตรง ไม่สามารถตีความหมายให้แน่ชัด ว่ามีสภาพการเรียนรู้มากน้อยเท่าไร เป็นตัวเลขลอย ๆ ไม่มีความหมาย เช่น นายจงรักสอบได้คะแนน 20 คะแนน นายจงดีสอบได้คะแนน 80 คะแนน ซึ่งคะแนน 230 และ 80 ถือว่าเป็นคะแนนดิบ

ธรรมชาติของคะแนนดิบมีลักษณะดังนี้

1) คะแนนดิบไม่ได้เป็นตัวเลขที่แสดงปริมาณความรู้แต่อย่างใด เช่นการที่ผู้เรียนคนหนึ่งสอบได้คะแนน 50 คะแนน จากคะแนนเต็ม 100 คะแนน แสดงว่าสอบได้คะแนนครึ่งหนึ่ง ของคะแนนเต็มเท่านั้น จะตัดสินว่าผู้เรียนมีความรู้ความสามารถครึ่งหนึ่งของวิชานั้นย่อมไม่ได้ เพราะขอบเขตของความรู้ในแต่ละวิชานั้นหาไม่ได้

2) คะแนนดิบไม่มีความหมายในตัวเอง แต่จะบอกเพียงว่าทำถูกเพียงใดเท่านั้น ไม่สามารถบอกได้ว่ามีความรู้มากหรือมีความรู้น้อย

3) คะแนนดิบจะมากหรือน้อยขึ้นกับความยากง่ายของข้อสอบ หากข้อสอบยากคะแนนของผู้สอบจะน้อย หากข้อสอบวิชาใดง่ายผู้สอบจะได้คะแนนมาก

4) คะแนนดิบแต่ละวิชาจะนำมาเปรียบเทียบกันไม่ได้ว่าใครได้คะแนนวิชาอะไรมากกว่าวิชาอะไร เพราะแต่ละวิชาคะแนนเต็มไม่เท่ากัน และหน่วยคะแนนมีขนาดไม่เท่ากัน

คะแนนแปลงรูป (Derived scores)

คะแนนแปลงรูป(Derived scores) คะแนนแปลงรูปเป็นคะแนนที่ได้จากการนำคะแนนดิบไปปรับเปลี่ยนด้วยกระบวนการทางคณิตศาสตร์ ให้เป็นรูปคะแนนใหม่ ที่มีความหมายสมบูรณ์ยิ่งขึ้น สามารถอ่านและเข้าใจได้ว่าผู้สอบที่ได้คะแนนนั้น ๆ เก่งหรืออ่อนเท่าไรโดยมีเกณฑ์เปรียบเทียบ

คะแนนแปลงรูปได้แก่คะแนนเปอร์เซ็นต์ไทล์และคะแนนมาตรฐาน

1) คะแนนเปอร์เซ็นต์ไทล์ เป็นคะแนนที่บอกให้รู้ว่าในหนึ่งร้อยคนมีคนที่ได้คะแนนต่ำกว่าหรือเท่ากับเขากี่คน เช่น P_{60} หมายความว่าใน 100 คนมีคนได้คะแนนต่ำกว่าหรือเท่ากับเขา 60 คน

การคำนวณหาเปอร์เซ็นต์ไทล์มีวิธีการดังนี้

- หาช่วงพิสัยของคะแนนโดยหาคะแนนต่ำสุดและสูงสุดในกลุ่มและเรียงคะแนนจากมากไปน้อยหรือน้อยไปมาก
- บันทึกความถี่ของแต่ละคะแนน
- หาความถี่สะสม
- หาค่าอันดับของเปอร์เซ็นต์ไทล์จากสูตร

$$P = \frac{\left(cf + \frac{f}{2}\right) \times 100}{N}$$

เมื่อ cf = ความถี่ในชั้นที่ต่ำลงมา

f = ความถี่ในชั้นที่ต้องการหา

N = จำนวนผู้เข้าสอบ

2) **คะแนนมาตรฐาน**(Standard scores)

คะแนนมาตรฐาน (Standard scores) คะแนนมาตรฐานเป็นการนำคะแนนดิบที่ได้ ไปเทียบกับคะแนนเฉลี่ยของกลุ่ม โดยพิจารณาว่ามากกว่าหรือน้อยกว่า คะแนนเฉลี่ยของกลุ่มเท่าไร คะแนนมาตรฐานที่นิยมใช้ได้แก่ คะแนน Z และคะแนน T

2.1) **คะแนน** Z (Z – score) หมายถึงผลต่างระหว่างคะแนนดิบกับคะแนนเฉลี่ยในหนึ่งส่วนเบี่ยงเบนมาตรฐาน เขียนเป็นสูตรได้

$$Z = \frac{X - \overline{X}}{S}$$

เมื่อ Z = คะแนนมาตรฐานของแต่ละคน

X = คะแนนดิบของแต่ละคน

$\overline{X}$ = คะแนนเฉลี่ยของกลุ่ม

S = ส่วนเบี่ยงเบนมาตรฐานของกลุ่ม

คุณลักษณะของคะแนน Z

- สามารถแปลความหมายได้โดยตัวของมันเองอย่างยุติธรรม
- สามารถนำมารวมกันได้ตามหลักคณิตศาสตร์
- คะแนนของผู้เรียนกลุ่มเดียวกันสามารถนำมาเปรียบเทียบเทียบระดับความเก่งอ่อนกันได้ในต่างวิชากัน

ข้อจำกัดของคะแนน Z คือคะแนน มีทั้งค่าบวกและลบ การนำค่าดังกล่าวอาจผิดพลาดได้ง่าย เช่นลืมเครื่องหมายลบ ซึ่งคะแนนที่เป็นลบทำให้เกิดความยุ่งยากและอาจผิดพลาดในการคำนวณ

2.2) **คะแนน** T (T-score) เป็นการปรับคะแนน Z เพื่อไม่ให้คะแนนมาตรฐานที่ได้มีค่าเป็นลบ คะแนน T มีคุณสมบัติเหมือนคะแนน Z ทุกประการ มีสูตรดังนี้

$$T = 10Z + 50$$

เมื่อ T = คะแนนมาตรฐาน T

Z = คะแนนมาตรฐาน Z

คะแนน T มีคะแนนเฉลี่ยเป็น 50 และส่วนเบี่ยงเบนมาตรฐานเป็น 10 ส่วนคะแนน Z มีค่าเฉลี่ยเป็น 0 และส่วนเบี่ยงเบนมาตรฐานเป็น 1

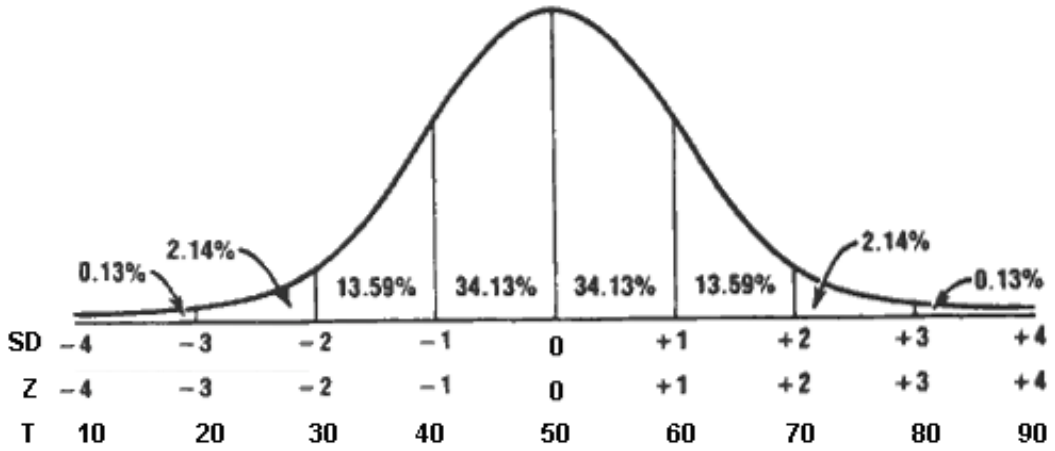
2.3.2 คะแนน T ปกติ (Normalized T-score)

การแปลงคะแนนดิบให้เป็นคะแนน T (T-score) เรียกว่าการแปลงเชิงเส้นตรง(Linear transformation) ซึ่งลักษณะการแจกแจงยังคงเหมือนคะแนนดิบ ดังนั้นจึงมีวิธีการแปลงคะแนนดิบ ให้เป็นคะแนนมาตรฐาน โดยยึดพื้นที่ใต้โค้งปกติ(area transformation) คือทำให้รูปโค้งการแจกแจงเปลี่ยนไปเข้าสู่รูปโค้งปกติมากยิ่งขึ้น คะแนนมาตรฐานที่ได้จากวิธีการแบบนี้เรียกว่าคะแนน T ปกติ

การแปลงคะแนนดิบให้เป็นคะแนน T ปกติ จะคำนวณโดยอาศัยพื้นที่ใต้โค้งปกติเป็นหลัก (Normal curve) โดยถือว่าพื้นที่ใต้โค้งปกติดังกล่าวจะใช้แทนจำนวนคนในกลุ่มที่เข้าสอบ

คุณสมบัติของโค้งปกติ

- เป็นรูปโค้งแบบระฆังคว่ำ โดยส่วนสูงของโค้งจะขึ้นอยู่กับค่าความแปรปรวน ถ้าข้อมูลมีความแปรปรวนน้อย โค้งจะสูงและฐานแคบ
- โค้งมีลักษณะสมมาตร
- ค่าเฉลี่ย มัธยฐาน และฐานนิยม จะมีค่าเท่ากัน และอยู่ตรงกึ่งกลางของโค้ง
- จุดสูงสุดของโค้งมีเพียงจุดเดียวคือจุดที่อยู่ยอดโค้ง
- ปลายโค้งทั้งสองข้างจะค่อย ๆ ลดต่ำลง แต่ไม่จรดแกนนอน
- พื้นที่ใต้โค้งที่อยู่ระหว่างส่วนเบี่ยงเบนมาตรฐาน +1 และ-1 จากค่าเฉลี่ย มีค่าเท่ากับ 68.26 % ระหว่างส่วนเบี่ยงเบนมาตรฐาน +2 และ -2 จากค่าเฉลี่ยมีพื้นที่เท่ากับ 95.44 % และระหว่างส่วนเบี่ยงเบนมาตรฐาน +3 และ -3 จากค่าเฉลี่ยมีพื้นที่เท่ากับ 99.74 %
- โค้งปกติที่ใช้กันอยู่ทั่วไป มีชื่อเรียกว่า Standard normal distribution ซึ่งมีค่าเฉลี่ยเท่ากับ 0 และส่วนเบี่ยงเบนมาตรฐานมีค่าเท่ากับ 1



รูป 2.1 แสดงการกระจายคะแนนแบบโค้งปกติ

ประโยชน์ของคะแนน T ปกติ

-กรณีที่มีการสอบ 1 วิชา แต่คะแนนแยกออกเป็น 2 ส่วนที่มีลักษณะต่างกัน เช่นคะแนนภาคปฏิบัติกับคะแนนภาคทฤษฎี ควรแปลงคะแนนดิบแต่ละส่วนให้เป็นคะแนน T ปกติ แล้วนำมารวมกันจะเป็น 2T แม้จะกำหนดน้ำหนักคะแนนไม่เท่ากันก็สามารถทำได้

- กรณีที่มีการสอบตั้งแต่ 2 วิชาขึ้นไป ถ้าต้องการรวมคะแนนเหล่านั้น ก็ต้องแปลงคะแนนดิบแต่ละวิชาให้เป็นคะแนน T ปกติ แล้วนำคะแนน T ปกติ ของแต่ละวิชามารวมกัน

- นำไปใช้ในการทำเกณฑ์ปกติ ซึ่งเป็นส่วนประกอบสำคัญของแบบทดสอบมาตรฐาน ใช้สำหรับตีความหมายคะแนนที่ได้จากการสอบ

- นำคะแนน T ปกติไปใช้ในการตัดสินผลการเรียน (ตัดเกรด) ในระบบอิงกลุ่มได้อย่างมีประสิทธิภาพ

2.3.3 **การกำหนดระดับคะแนนหรือตัดเกรด** (Grading)

การตัดเกรด(Grading) หรือการกำหนดระดับคะแนนเป็นสิ่งที่สำคัญที่สุดของกระบวนการวัดผลและประเมินผล ถ้าหากการทดสอบ การวัดผลและประเมินผล มีความถูกต้อง เที่ยงตรงทุกขั้นตอน การกำหนดระดับคะแนนก็จะต้องมีความถูกต้องและเที่ยงตรงด้วย ไม่เช่นนั้นจะเกิดผลเสียหายต่อกระบวนการวัดผลและประเมินผล

การตัดเกรดเป็นการสรุปผลการเรียนขั้นสุดท้าย โดยกำหนดระดับความสามารถในการเรียนของผู้เรียนว่า ผ่าน-ไม่ผ่าน หรือ เก่ง-อ่อน ระดับใด โดยสรุปออกมาเป็นระดับผลการเรียนหรือเกรด ซึ่งผู้สอนจะต้องพิจารณาอย่างรอบคอบ

องค์ประกอบของการตัดเกรด

- ผลการวัด การวัดที่ดีจะต้องให้ผลการวัดที่ถูกต้องแม่นยำ เที่ยงตรง ครอบคลุมและเชื่อถือได้

- เกณฑ์การพิจารณาต้องเป็นมาตรฐานที่ใช้เป็นหลักเปรียบเทียบหรือเป็นคุณลักษณะที่ตั้งไว้เป็นเป้าหมาย หรือมุ่งหวังที่จะให้เกิดแก่ผู้เรียนและใช้เป็นเครื่องตัดสิน ชี้ขาดระดับความสามารถของผู้เรียน

วิจารณญาณและคุณธรรม การประเมินผลที่เที่ยงตรง จำเป็นต้องอาศัยดุลยพินิจอย่างถี่ถ้วน รอบคอบ โดยพยายามให้เกิดความเป็นธรรม ขจัดความลำเอียงหรืออคติส่วนตัวออกไป และควรคำนึงถึงความเปลี่ยนแปลงงอกงามของผู้เรียนในด้านอื่น ๆ ประกอบด้วย

ขั้นตอนในการตัดเกรด

- กำหนดวัตถุประสงค์ของการสอบวัด ให้ชัดเจน
- สร้างเครื่องมือวัดตามจุดประสงค์
- กำหนดเกณฑ์การประเมิน
- ทดสอบผู้เรียน
- เปรียบเทียบผลกับเกณฑ์
- ประเมินความสามารถเป็นระดับคะแนน

ระบบการตัดเกรด มี 2 ระบบใหญ่ ๆ คือ

การตัดเกรดในระบบอิงเกณฑ์ (Criterion referenced)

วิธีนี้จะนำคะแนนที่เกิดจากผลการเรียนของผู้เรียนแต่ละคนไปเปรียบเทียบกับเกณฑ์ หรือจุดประสงค์ที่กำหนดไว้ การประเมินผลออกมาในรูปผ่านหรือไม่ผ่าน รู้หรือไม่รู้ หรือออกมาในรูปของเกรด A, B, C, D หรือ F เกรดที่ได้จากการประเมินแบบนี้ จะแปลความหมายว่าคนที่ได้เกรดใด สามารถทำอะไรได้บ้าง เช่น

A หมายถึงดีมาก (Excellent) ผู้ที่ได้จะต้องมีผลสัมฤทธิ์ทางการเรียนตามจุดมุ่งหมายสูงสุด ให้ระดับคะแนนเป็น 4

B หมายถึงดี (High satisfaction) ผู้ที่ได้จะต้องมีผลสัมฤทธิ์ทางการเรียนตามจุดมุ่งหมายเป็นส่วนมาก ให้ระดับคะแนนเป็น 3

C หมายถึงปานกลาง (Satisfaction) ผู้ที่ได้จะต้องมีผลสัมฤทธิ์ทางการเรียนตามจุดมุ่งหมายปานกลาง ให้ระดับคะแนนเป็น 2

D หมายถึงอ่อน (Poor) ผู้ที่ได้จะต้องมีผลสัมฤทธิ์ทางการเรียนตามจุดมุ่งหมายน้อยให้ระดับคะแนนเป็น 1

F หมายถึงอ่อนมาก (Unsatisfaction) ผู้ที่ได้จะต้องมีผลสัมฤทธิ์ทางการเรียนตามจุดมุ่งหมายน้อยมาก ให้ระดับคะแนนเป็น 0

ข้อดีของระบบนี้คือ สามารถจำแนกผู้เรียนได้เป็นกลุ่ม ๆ ตามระดับความสามารถทำให้เป็นแรงจูงใจสำหรับผู้เรียน และเข้าใจง่าย

ข้อเสีย คือระดับคะแนนอาจไม่สามารถบ่งชี้ผู้เรียนว่าบรรลุผลเท่าไร ทำให้ผู้ที่ผลการเรียนอ่อนมีความวิตกกังวล นอกจากนี้ความหมายของคะแนนร้อยละไม่ชัดเจน ตลอดจนการกำหนดระดับคะแนนเป็นระดับโดยอาศัยคะแนนร้อยละขาดเกณฑ์ที่น่าเชื่อถือ

การตัดเกรดในระบบอิงกลุ่ม (Norm referenced)

วิธีการนี้จะนำคะแนนที่เกิดจากผลการเรียนของผู้เรียนแต่ละคนไปเปรียบเทียบกับคะแนนของกลุ่มในวิชาเดียวกัน การประเมินวิธีนี้จะรายงานออกมาในรูปอันดับที่ หรือตำแหน่งเปอร์เซ็นต์ไทล์ หรือในรูปเกรด A, B, C, D หรือ F เกรดในการประเมินแบบนี้จะให้ความสำคัญในลักษณะที่บ่งบอกว่าเขาเก่งกว่าคนอื่นเท่าไร ไม่สามารถบอกได้ว่าเกรดนั้น เขามีความรู้ในเรื่องใดบ้างเป็นจำนวนเท่าใด

ข้อดีของการตัดเกรดระบบอิงกลุ่มคือเข้าใจง่าย เป็นที่ยอมรับกันในปัจจุบัน

ข้อเสีย คะแนนที่ได้จากแบบทดสอบไม่ค่อยเป็นมาตรฐาน ถ้าผู้สอบมีจำนวนน้อยทำให้โอกาสการกระจายคะแนนได้น้อย ดังนั้นจะใช้ได้ดีกับผู้เรียนจำนวนมากหลายพันคน ผู้เรียนมีการแข่งขันมาก เกิดความเครียดมาก ถ้าเป็นชั้นเรียนที่ที่มีผู้เรียนที่คัดสรรแล้ว ทำให้คะแนนที่ได้จากผู้เรียนมีโอกาสกระจายเป็นโค้งปกติยาก

การตัดเกรดในระบบอิงกลุ่มมีหลายแบบ เช่นจัดกลุ่มตามธรรมชาติ การกำหนดเปอร์เซ็นต์ตามการแจกแจงของโค้งปกติ วิธีของดักลาส วิธีของสตุท การใช้ค่ามัธยฐาน การใช้คะแนนมาตรฐาน Z การใช้คะแนนมาตรฐาน T ปกติ การใช้คะแนนมาตรฐานเก้า

ในที่นี้จะกล่าวถึงแบบการใช้คะแนนมาตรฐาน T ปกติ ซึ่งคะแนนกระจายอยู่ในรูปของโค้งปกติ และจำนวนเกรดขึ้นกับดุลยพินิจของผู้ประเมิน

ตัวอย่างจากการสอบวิชาฟิสิกส์ ผู้สอบจำนวน 25 คน ได้คะแนนสอบและแปลงเป็นคะแนน T ปกติ ได้ดังตาราง

คะแนน	ความถี่	คะแนน T ปกติ
23	1	71
22	2	64
21	2	60
20	3	56
19	4	53
18	2	49
17	4	46
16	2	43
15	0	42
14	2	40
13	2	36
12	1	29

ขั้นตอนในการตัดเกรดมีดังนี้

- หาพิสัยของคะแนน T ปกติ = 71-29 = 42

- พิจารณาเกรดที่ต้องการ อาจจะเป็น 2, 3, 4หรือ 5 เกรด

- นำจำนวนเกรดที่ต้องการไปหารค่าพิสัย ผลลัพธ์ที่ได้คืออันตรภาคชั้นหรือคะแนนของแต่ละเกรด เช่นต้องการตัด 2 เกรด ใช้คะแนน T ปกติที่ 50 เป็นหลัก แล้วแบ่งคะแนน T ปกติที่สูงกว่า 50 ได้เกรดหนึ่ง และที่คะแนน T ปกติต่ำกว่า 50 เป็นอีกเกรดหนึ่ง ซึ่งเราจะกำหนดเป็นเกรดอะไรก็ได้ เช่นถ้ากำหนดเกรด A และ B จะได้ว่ามีผู้ได้เกรด A จำนวน 12 คน และได้เกรด B จำนวน 13 คน

ถ้าต้องการตัดเกรด 3 เกรด เช่น A, B และ C หาจำนวนคะแนนในแต่ละเกรด = $\frac{42}{3}$ = 14 และ $\frac{14}{2}$ = 7

ดังนั้น เกรด A คือคะแนน T ตั้งแต่ 57 ขึ้นไป มีจำนวน 5 คน

เกรด B คือคะแนน T ตั้งแต่ 44-56 มีจำนวน 13 คน

เกรด C คือคะแนน T ตั้งแต่ 43 ลงมา มีจำนวน 7 คน

ถ้าต้องการตัดเกรด 4 เกรด เช่น A, B, C และ D หาจำนวนคะแนนในแต่ละเกรด = $\frac{42}{4}$ = 10.5

ดังนั้น เกรด A คือคะแนน T ตั้งแต่ 61 ขึ้นไป มีจำนวน 3 คน
เกรด B คือคะแนน T ตั้งแต่ 51-60 มีจำนวน 9 คน
เกรด C คือคะแนน T ตั้งแต่ 40-50 มีจำนวน 10 คน
เกรด D คือคะแนน T ตั้งแต่ 39 ลงมา มีจำนวน 3 คน

ถ้าต้องการตัดเกรด 5 เกรด เช่น A, B, C, D และ F หาจำนวนคะแนนในแต่ละเกรด = $\frac{42}{5}$ = 8.4 และ $\frac{8.4}{2}$ = 4.2

ดังนั้น เกรด A คือคะแนน T ตั้งแต่ 63 ขึ้นไป มีจำนวน 3 คน
เกรด B คือคะแนน T ตั้งแต่ 55-62 มีจำนวน 5 คน
เกรด C คือคะแนน T ตั้งแต่ 46-54 มีจำนวน 10 คน
เกรด D คือคะแนน T ตั้งแต่ 38-45 มีจำนวน 4 คน
เกรด F คือคะแนน T ตั้งแต่ 37 ลงมา มีจำนวน 3 คน

หมายเหตุ จำนวนเกรดที่ต้องการ เช่น 3 เกรด ไม่จำเป็นต้องเป็น A, Bและ C อาจจะเป็น B, C และ D ก็ได้ ไม่ว่าจะตัดกี่เกรด ควรเริ่มต้นแบ่งเกรดจากคะแนน T ที่ 50 เสมอ

บทที่ 3
วิธีการดำเนินการวิจัย

การพัฒนาโปรแกรมการคำนวณคะแนน T ปกติ เพื่อกำหนดระดับคะแนนผ่านเครือข่ายอินเตอร์เน็ต เป็นการเขียนโปรแกรมโดยใช้ภาษาจาวาทั้งหมด (Pure Java) ไม่มีการใช้ภาษา Script อื่น ๆ เข้าช่วย เครื่องมือที่ใช้ในการเขียนโปรแกรม มีดังนี้

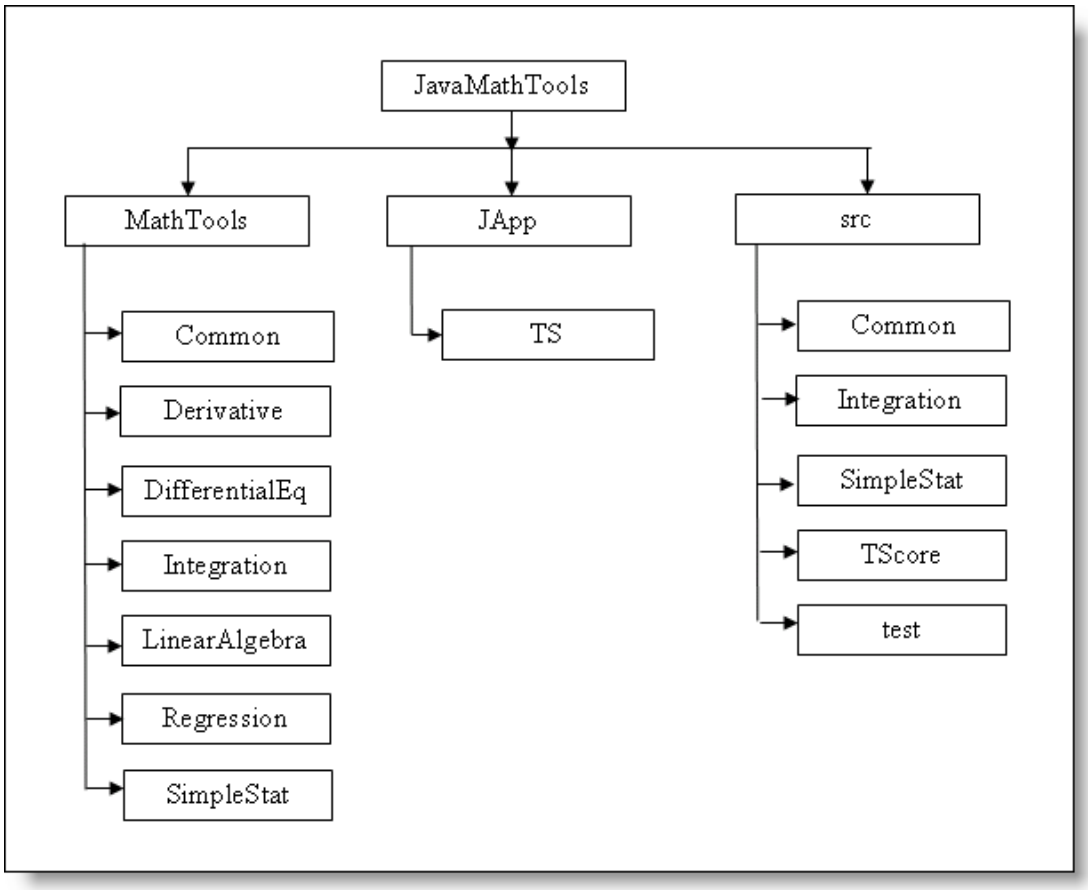
1. เครื่องคอมพิวเตอร์สำหรับใช้เขียนโปรแกรม และทดสอบโปรแกรม
2. ซอฟต์แวร์ Java SDK 1.6 (Java Standard Development Kit) สามารถ download ได้โดยไม่เสียค่าใช้จ่ายจาก http:java./sun.com
3. ซอฟต์แวร์ ที่ใช้เป็น Editor สำหรับเขียนโปรแกรมต้นฉบับ หรือ Source code ส่วนใหญ่จะเป็น Open source สามารถ download มาใช้ได้โดยไม่เสียค่าใช้จ่าย ได้แก่ Netbean, Eclipse, Jcreator light หรือ Notepad ที่ติดมาพร้อมกับระบบปฏิบัติการวินโดว์
4. เครื่องคอมพิวเตอร์ทำหน้าที่เป็น server หมายเลข IP คือ 203.158.100.140 ใช้โปรแกรม Apache 2 เป็น web server ภายใต้ระบบปฏิบัติการวินโดว์ 2003 สำหรับทดสอบการทำงานของโปรแกรมผ่านเครือข่ายอินเตอร์เน็ต ในกรณีที่ไม่สามารถเชื่อมต่ออินเตอร์เน็ต อาจจะจำลองคอมพิวเตอร์เครื่องใด ๆ ให้เป็น web server ก็ได้ โดยใช้ เลข IP 127.0.0.1

การวิจัยนี้ได้รับงบประมาณแผ่นดินประจำปี. 2551 การใช้เงินงบประมาณจะเริ่มตั้งแต่ตุลาคม 25450 สิ้นสุด กันยายน 2551 ผู้วิจัยได้วางแผนการพัฒนา โดยมีแผนงานเป็นลำดับดังนี้

ขั้นตอนที่ 1 ศึกษาค้นคว้าการคำนวณหาคะแนน Normalized T Score เริ่มตั้งแต่การนำคะแนนดิบมาจัดเรียงลำดับจากมากไปน้อย การคำนวณหาความถี่สะสม แล้วนำไปคำนวณหาพื้นที่ใต้เส้นโค้งปกติ ได้ผลลัพธ์เป็นค่า Normalized T Score จากนั้นจึงออกแบบระเบียบวิธีการคำนวณ ได้แก่ การจัดลำดับคะแนน การหาความถี่ของคะแนนการหาความถี่สะสม การคำนวณปริพันธ์หาพื้นที่ใต้โค้งปกติ โดยออกแบบให้ใช้เวลาในการประมวลผลน้อยที่สุด ใช้ชุดคำสั่งรวมทั้งใช้ทรัพยากรของคอมพิวเตอร์ของเครื่องน้อยที่สุด

ขั้นตอนที่ 2 เขียนคำสั่งโปรแกรมเป็นภาษาจาวา ทำการคอมไพล์ให้เป็น byte code อยู่ในรูปของไฟล์ class คำสั่งต่าง ๆ ที่เขียนในโปรแกรมต้นฉบับนี้จะยึดมาตรฐานของจาวารุ่นที่ 6.0 หรือทันสมัยกว่าเป็นหลัก นั่นหมายถึงการนำโปรแกรมต้นฉบับไปคอมไพล์โดยใช้จาวาที่มีรุ่นต่ำกว่า 6.0 อาจพบข้อความผิดพลาดขณะทำการคอมไพล์ อาจจะต้องดัดแปลงหรือแก้ไขคำสั่งบางคำสั่งให้สอดคล้องกับจาวารุ่นที่เลขเวอร์ชันต่ำกว่า 6.0

เพื่อความเป็นระเบียบและเป็นระบบในการเขียนโปรแกรม และเพื่อความสะดวกในการนำไปใช้งาน ผู้วิจัยจึงได้จัดสร้างโฟลเดอร์ มีลักษณะเป็นลำดับชั้นดังนี้



ในโฟลเดอร์ MathTools จะเก็บ class file ที่ผ่านการคอมไพล์ แบ่งเป็นโฟลเดอร์ย่อยตามลักษณะหัวข้อทางคณิตศาสตร์

โฟลเดอร์ Common คลาสในโฟลเดอร์นี้ใช้เก็บค่าคงที่ทั่วไป ค่าคงที่ทางคณิตศาสตร์และฟิสิกส์ ฟังก์ชันที่จะต้องใช้ร่วมกันทั้งproject

โฟลเดอร์ Derivative เก็บคลาสที่ใช้แก้ปัญหาการหาอนุพันธ์

โฟลเดอร์ DifferentialEq เก็บคลาสใช้ในการแก้สมการอนุพันธ์แบบสามัญ

โฟลเดอร์ Integration เก็บคลาสการหาปริพันธ์เบื้องต้น

โฟลเดอร์ LinearAlgebra เก็บคลาสการคำนวณเมตริกซ์ และดิเทอร์มิแนนท์ การแก้สมการระบบสมการเชิงเส้น

โฟลเดอร์ Regression เก็บคลาสที่ใช้การประมาณค่าโดยวิธีกำลังสองน้อยที่สุด ได้แก่ Linear regression และ Polynomial regression.

โฟลเดอร์ SimpleStat เก็บคลาสที่ใช้ในการคำนวณทางสถิติเบื้องต้น ได้แก่ การหาความถี่ ความถี่สะสมของข้อมูล การหาค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐาน การคำนวณหาค่า Z รวมทั้งการคำนวณคะแนน T ปกติ

ในโฟลเดอร์ JApp เป็นโฟลเดอร์ที่ใช้เก็บ โปรแกรมประเภท Application ซึ่งเรียกใช้คลาสต่าง ๆ ที่อยู่ใน โฟลเดอร์ MathTools แบ่งเป็นโฟลเดอร์ย่อย ตามชื่อ Application ที่สร้างขึ้น ซึ่งขณะนี้มีเพียงโฟลเดอร์ย่อยเพียงโฟลเดอร์ย่อยเพียงโฟลเดอร์เดียวคือ TS (เป็นชื่อย่อของ TScore) ในอนาคตถ้าต้องการสร้าง Application อื่น ๆ อีก ก็สามารถเก็บเป็นโฟลเดอร์ย่อยภายใต้โฟลเดอร์ JApp นี้

ในโฟลเดอร์ src ใช้เก็บโปรแกรมต้นฉบับ (Source code) ในที่นี้จะแสดงเฉพาะโฟลเดอร์ของโปรแกรมต้นฉบับที่เกี่ยวข้องกับโปรแกรมคำนวณคะแนน T ปกติ เพื่อกำหนดระดับคะแนนผ่านเครือข่ายอินเตอร์เน็ต เท่านั้น ซึ่งมีโฟลเดอร์ย่อยที่เกี่ยวข้อง 5 โฟลเดอร์ย่อย คือ

โฟลเดอร์ Common เป็นโปรแกรมต้นฉบับที่เก็บค่าคงที่ ซึ่งโปรแกรมคำนวณคะแนน T ปกติ ฯ นี้มีการเรียกใช้ค่าคงที่เหล่านี้

โฟลเดอร์ Integration เก็บโปรแกรมต้นฉบับที่ใช้หาค่าปริพันธ์แบบต่าง ๆ

โฟลเดอร์ SimpleStat เก็บโปรแกรมต้นฉบับที่ใช้คำนวณค่าสถิติต่าง ๆ ที่เกี่ยวข้องกับโปรแกรม

โฟลเดอร์ TScore เก็บโปรแกรมต้นฉบับที่ใช้คำนวณ คะแนน T และ การตรวจสอบการรับข้อมูลจากแป้นพิมพ์ หรือจากไฟล์ ก่อนที่จะนำไปสู่ขั้นตอนการคำนวณ เมื่อคอมไพล์ได้แล้วจะถูกเก็บรวมไว้ในโฟลเดอร์ MathTools/SimpleStat

โฟลเดอร์ test เก็บโปรแกรมต้นฉบับ ของ Application แบ่งตามเวอร์ชันต่าง ๆ ที่พัฒนามาเป็นลำดับ รวมทั้งส่วนที่เป็น applet และส่วนที่เป็น Desktop application

ขั้นตอนที่ 3 ทดสอบการทำงานของโปรแกรม โดยนำคะแนนจากผลการสอบของแต่ละภาคการศึกษามาทดสอบ ตรวจสอบผลลัพธ์จากโปรแกรมโดยเปรียบเทียบกับผลลัพธ์ที่ได้วิธีการคำนวณโดยการเปิดตาราง ตรวจสอบค่า Z ที่คำนวณได้จากพื้นที่ใต้เส้นโค้งกับตารางแสดงค่า Z ที่ปรากฏอยู่ท้ายหนังสือเกี่ยวกับสถิติ และตรวจสอบค่าที่ได้กับโปรแกรมสำเร็จรูป Excel

ขั้นตอนที่ 4 ปรับปรุงหรือแก้ไขชุดคำสั่ง ในกรณีที่ประมวลผลแล้วได้ค่าคลาดเคลื่อนหรือมีการใช้เวลาในการประมวลผลนานผิดปกติ

ถ้าพบว่ามีการประมวลผลแล้วได้คำตอบคลาดเคลื่อนไปจากค่าแท้จริงมาก ผู้วิจัยต้องเริ่มตรวจสอบตั้งแต่ชุดคำสั่งที่ถอดมาจากระเบียบวิธีการคำนวณที่ออกแบบไว้ ถ้าชุดคำสั่งไม่มีข้อผิดพลาด ต้องย้อนกลับไปดูระเบียบวิธีการคำนวณแต่ละขั้นตอนว่าขั้นตอนใดทำให้เกิดข้อผิดพลาดบ้าง อาจต้องออกแบบหรือแก้ไขปรับปรุงระเบียบวิธีการคำนวณใหม่ เพื่อให้ได้คำตอบที่ใกล้ค่าแท้จริง เกิดความคลาดเคลื่อนน้อยที่สุด

เมื่อการประมวลผลของโปรแกรมได้ผลลัพธ์ตามจุดประสงค์แล้ว จึงเริ่มจัดทำคู่มือการใช้งานโปรแกรม ตั้งแต่การ download โปรแกรม การติดตั้งโปรแกรม การเริ่มใช้งานโปรแกรม ข้อควรระวังในการใช้ รวมทั้งวิธีแก้ไข กรณีที่เกิดข้อผิดพลาดเนื่องมาจากผู้ใช้งานไม่ถูกต้อง

ขั้นตอนที่ 5 นำไปติดตั้งไว้ในเครื่องแม่ข่ายเพื่อใช้งานผ่านทางอินเทอร์เน็ต ผู้ใช้สามารถเข้าถึงได้โดยตรงที่ http://203.158.100.140/TScore/

ผู้ใช้งานสามารถเลือกให้โปรแกรมทำงานอย่างใดอย่างหนึ่งใน 2 แบบ คือ ทำงานในลักษณะแอพเพล็ต ซึ่งโปรแกรมจะทำงานภายใต้บราวเซอร์ที่ผู้ใช้กำลังใช้งานอยู่ การใช้งานในลักษณะนี้ ผู้ใช้ไม่สามารถให้โปรแกรมอ่านไฟล์จากฮาร์ดดิสก์ หรือ พิมพ์ผลลัพธ์ผ่านทางเครื่องพิมพ์ได้ เพราะข้อกำหนดความปลอดภัยของการทำงานของแอพเพล็ตได้กำหนดเป็นข้อห้ามไว้ อีกวิธีหนึ่งคือ ผู้ใช้สามารถดาวน์โหลดโปรแกรม ให้มาทำงานที่เครื่องคอมพิวเตอร์ของผู้ใช้ โปรแกรมจะทำงานนอกบราวเซอร์ เสมือนหนึ่งเป็นโปรแกรมที่ถูกติดตั้งอยู่ภายในเครื่องคอมพิวเตอร์ของผู้ใช้เอง เมื่อสิ้นสุดการทำงานแล้ว ก็จะไม่เหลือโปรแกรมนี้อยู่ในตัวเครื่อง ทำให้ไม่สิ้นเปลืองเนื้อที่ติดตั้งโปรแกรม แต่ถ้าผู้ใช้มีความประสงค์ต้องการจะเก็บโปรแกรมนี้ไว้ใช้งานอีกก็สามารถ Save โปรแกรมนี้เก็บไว้ในเครื่องได้ เมื่อเลิกเชื่อมต่ออินเตอร์เน็ตแล้วก็ยังคงสามารถใช้โปรแกรมนี้ได้เหมือนตอนที่กำลังเชื่อมต่ออินเตอร์เน็ต

บทที่ 4
ผลการวิจัย

ในการพัฒนาโปรแกรมการคำนวณคะแนน T ปกติ เพื่อกำหนดระดับคะแนนผ่านเครือข่ายอินเตอร์เน็ต ผู้วิจัยได้เขียนด้วยภาษาคอมพิวเตอร์ที่เรียกว่า Java โดยอาศัยการเขียนโปรแกรมในลักษณะเชิงวัตถุ (Object Oriented Programming, OOP) ออกแบบเป็นคลาสต่าง ๆ แล้วนำคลาสเหล่านั้นมาทำงานร่วมกัน เมื่อโปรแกรมต้นฉบับผ่านการคอมไพล์เป็นไฟล์ class แล้ว จะได้โปรแกรมที่อยู่ในรูป Byte Code เล็ก ๆ หลาย ๆ ไฟล์ ซึ่งเหมาะแก่การขนย้ายผ่านเครือข่ายอินเตอร์เน็ต พบว่าขนาดของไฟล์ที่สำเร็จแล้วและใช้งานได้ ทั้งในกรณีที่ทำงานแบบแอพเพล็ต และทำงานแบบ Application ขนาดของไฟล์หลังจากผ่านการบีบอัดเป็น Jar file แล้วไม่เกิน 40 kB ทั้งสองแบบ

จะกล่าวถึงการพัฒนาโปรแกรมโดยแบ่งตัวโปรแกรมออกเป็นส่วนย่อยดังนี้

1. ส่วนที่เป็น Input ของข้อมูล
2. ส่วนที่คำนวณค่าสถิติพื้นฐาน
3. ส่วนที่คำนวณคะแนน ที ปกติ และเลือกพิสัยของระดับคะแนน
4. ส่วนที่แสดงผลลัพธ์
5. ส่วนที่แสดงข้อความแนะนำการใช้งานโปรแกรม
6. การนำโปรแกรมไปติดตั้งที่ แม่ข่ายอินเตอร์เน็ต
7. ผลการทำงานของโปรแกรมผ่านเครือข่ายอินเตอร์เน็ต

4.1 ส่วนที่เป็น Input ของข้อมูล

โปรแกรมนี้สามารถรับข้อมูลจากการพิมพ์คะแนนดิบลงไปในกรอบป้อนข้อมูล (Text area) ของโปรแกรม แต่ถ้าให้โปรแกรมนี้ทำงานแบบ Desktop application สามารถรับข้อมูลได้จากการอ่านค่าจาก Text file ได้อีกทางหนึ่ง

การพิมพ์คะแนนดิบลงไปใน Text area หรือ เก็บไว้เป็น Text file มีหลักเกณฑ์เช่นเดียวกันดังนี้

-คะแนนดิบจะเป็นจำนวนเต็ม หรือ จำนวนจริง ก็ได้

-คั่นคะแนนดิบแต่ละค่าด้วยช่องว่าง (เกิดจากการเคาะปุ่ม Spacebar) ห้ามใช้เครื่องหมายจุลภาค หรือใช้ tab คั่น

-ข้อมูลคะแนนดิบจึงเป็นตัวเลขและจุดทศนิยมเท่านั้น ห้ามพิมพ์สัญลักษณ์ หรืออักขระใด ๆ

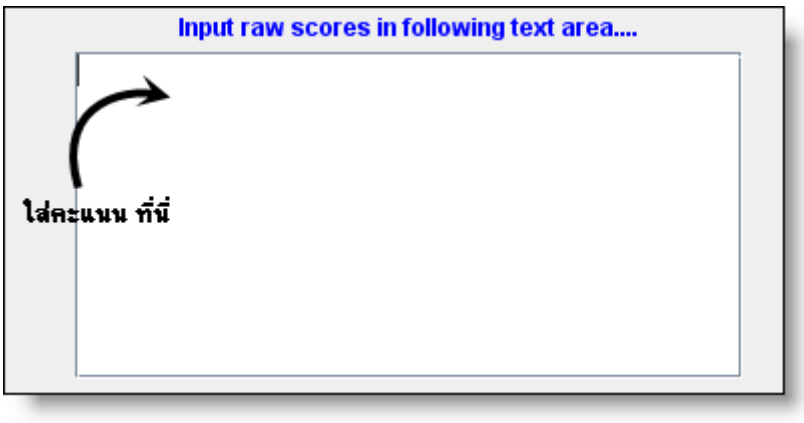
-ถ้าต้องการพิมพ์หมายเหตุ หรือ Comment สำหรับไว้เป็นบันทึกช่วยจำ ให้ใส่เครื่องหมายดาว (*) ไว้ที่หน้าบรรทัดนั้น บรรทัดใดที่มีเครื่องหมาย * นำหน้า โปรแกรมจะข้ามไปไม่นำไปประมวลผล

- ผู้ใช้สามารถพิมพ์ข้อมูลลงใน Editor ใด ๆ แล้วใช้วิธี Copy และ Paste ข้อมูลลงมาใน Text area

-ถ้าผู้ใช้ป้อนข้อมูลผิดพลาด เช่น มีจุดทศนิยมมากกว่า 1 จุดในค่าเดียวกัน หรือมีตัวอักษรปนในข้อมูล โปรแกรมจะแสดงข้อความผิดพลาด และบอกบรรทัดที่เกิดข้อผิดพลาดนั้น ผู้ใช้จะต้องแก้ไขให้ถูกต้องเสียก่อน โปรแกรมจึงจะทำงานต่อไป

คลาสที่ทำหน้าที่รับข้อมูลจากผู้ใช้ คือคลาส ScoreTextArea โดยออกแบบให้คลาสนี้สืบทอดมาจากคลาส JTextArea ซึ่งเป็นคลาสที่มีอยู่แล้วใน API ของภาษาจาวา ผู้พัฒนาได้เพิ่มคุณสมบัติและความสามารถเพิ่มเติมคือ สามารถตรวจสอบข้อมูลที่เป็นจำนวนเต็มและจำนวนจริง และนำข้อมูลที่ผ่านการตรวจสอบแล้วไปเก็บไว้ในอะเรย์ (Array) ที่ชื่อว่า Score ซึ่งมีขนาดเท่ากับ 2000 (เท่ากับจำนวนข้อมูลที่สามารถเก็บได้สูงสุด) เพื่อความสะดวกในการขยายขอบเขตของข้อมูลในอนาคต จึงได้สร้างคลาสชื่อ TScoreConstants เพื่อเก็บค่าคงที่ที่ใช้กับทุก ๆ คลาสในโปรแกรม จะพบตัวแปรชื่อ MAXIMUM_NUMBER_OF_DATA ซึ่งมีค่าเท่ากับ 2000 ถ้าต้องการขยายขีดความสามารถของโปรแกรมให้รองรับข้อมูลมากกว่านี้ สามารถทำได้ง่ายโดยการแก้ค่า MAXIMUM_NUMBER_OF_DATA ตรงคลาสนี้เพียงที่เดียว

เมื่อแสดงผลบนจอภาพ จะเห็นบริเวณที่เป็นกรอบป้อนข้อมูล ที่ผู้ใช้สามารถป้อนคะแนนดิบเข้าไปได้ ดังรูปที่ 4.1



รูปที่ 4.1 แสดงกรอบป้อนข้อมูลที่ผู้ใช้ใส่คะแนนดิบ

Listing 4.1 แสดงรายละเอียดของคลาส TScoreConstants

```
 1:  /* File:TScoreConstants.java
 2:  * Project :  Nomalized TScore Calculation
 3:  * Author  : Wachara R.
 4:  * First Released: 19 Jun 2008
 5:  * Last Updated :
 6:  */
 7:  package MathTools.SimpleStat;
 8:  public class  TScoreConstants
 9:  {
10:   /* Constants for calculating in All TScore programs  */
11:        public static final int MAXIMUM_NUMBER_OF_DATA = 2000;
12:
13:        public static final double DEFAULT_TOLERANCE  = 1.0e-7;
14:
15:  }
16:
```

Listing 4.2 แสดงรายละเอียดของคลาส ScoreTextArea

```
 1:  /* File : ScoreTextArea.java
 2:  * Project: T Score Calculation
 3:  * Purpose: Raw score input in  JTextArea  and then
 4:  *                converse them to number.
 5:  * Author : Wachara R.
 6:  * First Released: Sun 9 March 2008
 7:  * Last Updated : Tue 23 May 2008
 8:  */
 9:  package MathTools.SimpleStat;
10:
11:  import javax.swing.*;
12:  import java.awt.*;
13:  import java.awt.event.*;
14:  import static MathTools.SimpleStat.TScoreConstants.*;
15:
16:  public class  ScoreTextArea extends JTextArea
17:  {
18:  Font textFont = new Font("Dialog", Font.PLAIN, 12);
19:  int NumberOfScore;
20:  private double [ ] Score  = new double[MAXIMUM_NUMBER_OF_DATA];
21:
22:  public ScoreTextArea ()
23:        {
24:                    setFont(textFont);
25:                    setBackground(Color.white);
26:                    setSelectionColor(Color.orange);
27:                    setLineWrap(false);
          // Wrap the line at the container end.
28:        }
29:  public int readScore( )
30:        {
31:        String  score = super.getText();
32:      char[] ch = score.toCharArray();
33:        String temp="";
34:        boolean VALIDED=true;
35:        boolean IN_COMMENT=false;
36:        boolean  IN_DIGIT = false;
37:        int point = 0;
38:        int line = 0;
39:        int n = 0;
40:
```

```
41:             for(int i =0; i < ch.length ; i++)
42:       {
43:     switch (ch[i])
44:             {
45:             case '*' :
46:                   IN_COMMENT = true;
47:                   break;
48:             case '0':
49:             case '1':
50:             case '2':
51:             case '3':
52:             case '4':
53:             case '5':
54:             case '6':
55:             case '7':
56:             case '8':
57:             case '9':
58:             case '.':
59:
60:                   if (!IN_COMMENT)
61:             {
62:                   IN_DIGIT =true;
63:                   if (ch[i] == '.')  point +=1;
64:             temp = temp + ch[i];
65:                   if (point > 1)   {
66:   JOptionPane.showMessageDialog(null,"Some Score not
    (a)numbers atline"+(line+1),"Error",JOptionPane.ERROR_MESSAGE);
67:                   return -1;
68:                   }
69:             }
70:             break;
71:
72:             case ' ':
73:             case '\n':
74:                   if (ch[i] == '\n')
75:             {
76:                   line = line +1;
77:                   IN_COMMENT = false;
78:             }
79:             if (!(IN_COMMENT ) && IN_DIGIT)
80:             {
81:                   point = 0;
82:                   if (temp != "")
83:                   {
84:                         Score[n] = Double.parseDouble(temp);
85:                         temp = "";
86:                         n=n+1;
87:                   }
88:                   IN_DIGIT = false;
89:             }
90:             break;
91:             default :
92:                   if (!IN_COMMENT)
93:                   {
94:               JOptionPane.showMessageDialog(null,
         "Error at line"
    +(line+1),"Error",JOptionPane.ERROR_MESSAGE);
95:                         return -1;
96:                   }
97:                   break;
98:             } //switch
```

```
 99:
100:        if ( i == (ch.length-1) && ch[i] != '\n') {
101:                Score[n] = Double.parseDouble(temp);
102:                        n=n+1;
103:
104:          }
105:
106:                    }// for loop
107:                    return n;
108:        }
109:        public double [ ]  getScore()
110:        {
111:              return Score;
112:        }
113:  }
114:
```

การตรวจสอบข้อมูลคะแนนดิบว่าถูกต้องตามกฎเกณฑ์หรือไม่ โดยใช้เมธอด readScore() ตั้งแต่บรรทัดที่ 29 ถึงบรรทัด ที่ 99 ถ้าข้อมูลไม่ถูกต้องจะแสดงหน้าต่างแจ้งข้อความผิดพลาด ดังบรรทัดที่ 66 และ 94

คลาส ScoreTextArea นี้เมื่อผ่านการคอมไพล์แล้วจะถูกเก็บรวมอยู่ในโฟลเดอร์ SimpleStat

คลาส TScorePanel จะเป็นคลาสเรียกใช้ ScoreTextArea อีกทีหนึ่ง จะแสดง Text area ปรากฏบนจอภาพ รายละเอียดของคลาส TScorePanel จะกล่าวอย่างละเอียดในหัวข้อ 4.4 ส่วนที่แสดงผลลัพธ์

4.2 ส่วนที่คำนวณค่าสถิติพื้นฐาน

สถิติที่ใช้ในการคำนวณในโปรแกรมนี้ได้แก่ การหาค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐาน ความถี่ของข้อมูล ความถี่สะสม พื้นที่ใต้เส้นโค้งปกติ และค่า Z ที่ทำเกิดพื้นที่ใต้เส้นโค้ง ผู้พัฒนาได้สร้างคลาสเพื่อจัดการคำนวณค่าสถิติเหล่านี้ ชื่อ SimpleStat รายละเอียดของคลาสมีดังนี้

Listing 4.3 แสดงรายละเอียดของคลาส SimpleStat

```
 1:  /* File : Simplestat.java
 2:  * Project: Math Tools in Java
 3:  * Purpose: Compute basic statistics
 4:  * frequency, central tendency, measure of variation.
 5:  *                 nomal curve distribution.
 6:  * Author : Wachara R.
 7:  * First Released: Th 3 May 2007
 8:  * Last Updated : Tu 8 July 2008
 9:  *     add constructor SimpleStat ( double, int)
10:  */
11:
12:  package MathTools.SimpleStat;
13:
14:  import static java.lang.Math.*;
```

```
15:  import static java.lang.Double.*;
16:  import MathTools.SimpleStat.SimpleStatException;
17:  import static MathTools.Common.CommonConstants.*;
18:  import MathTools.Common.Function;
19:  import MathTools.Integration.*;
20:
21:  /** class for basic statistics algorithm:  requency,
22:  *     central tendency, measure of variation.
23:  *     some inference statistics*/
24:
25:  public  class SimpleStat {
26:
27:  /** data for computing the basic statistics */ double[ ] data;
28:  /** number of data */  int  numData;
29:  /** number of data after finding frequency */
                            int numDataAtLast;
30:  /** basic statistic quantities */
31:  /** mean value */  double mean;
32:  /** median value */ double median;
33:  /** range of data */ double range;
34:  /** standard deviation */ double sd;
35:  //sorted data */ double [ ] sData;
36:  /** managed data */  double [] mData;
37:  /** keeping repeated mode data */ double[] modeData;
38:  /** mode of data */ int mode;
39:  /** frequency of data set */ int [] frequency;
40:  /** cumulated frequency */ int [] cFrequency;
41:  /** checking data have been sorted */ boolean isSorted = false;
42:  /** checking data have been classified with frequencies */
43:  boolean isClassified = false;
44:/** checking data have been computed for cumulated frequencies */
45:  boolean havingCumulatedFreq=false;
46:  /** Constructor
47:  * @param dat data for computing basic statistics
48:  */
49:  /** Constructor */
50:  public SimpleStat( ) {
51:  }
52:
53:  public SimpleStat( double [ ] dat ){
54:        data = dat;
55:        numData = dat.length;
56:        findMean();
57:              findSD();
58:  //    sData = new double[ numData];
59:  //    sData = copyData(data,numData);
60:        quickSort( data, 0,numData-1,false);
61:        findMedian();
62:         findFrequency();
63:        findCumulativeFrequency();
64:        findMode();
65:        findRange();
66:
67:        /*
68:        System.out.println("Data frequency...: ");
69:              for(int i = 0; i < numDataAtLast; i++){
70:        System.out.println( mData[i] +"   " + frequency[i]);
71:              }
72:
73:        */
74:  }
```

```
 75:  public SimpleStat( double [ ] dat, int numberOfData ){
 76:        data = dat;
 77:        numData = numberOfData;
 78:        findMean();
 79:              findSD();
 80:        quickSort( data, 0,numData-1,false);
 81:        findMedian();
 82:         findFrequency();
 83:        findCumulativeFrequency();
 84:        findMode();
 85:        findRange();
 86:
 87:  }
 88:  /** copy data from current array to another array */
 89:  private double[ ] copyData(double[] itemToBeCopied,int nItem) {
 90:        double[ ] dummy  = new double[ nItem];
 91:        for(int i = 0 ; i < nItem; i++)
 92:              dummy[i] = itemToBeCopied[i];
 93:        return dummy;
 94:        }
 95:
 96:        private int[ ] copyData(int[] itemToBeCopied,int nItem) {
 97:        int[ ] dummy  = new int[ nItem];
 98:        for(int i = 0 ; i < nItem; i++)
 99:              dummy[i] = itemToBeCopied[i];
100:        return dummy;
101:        }
102:
103:
104:        // ========= Frequency ==========
105:        /** finding frequency of each data */
106:        private void  findFrequency(  ){
107:        int n_freq = 0;
108:        int tempfrequency[ ]  = new int[numData];
109:        double temp [ ]  = new double[numData];
110:        double dummy[] = new double [numData];
111:        if (isSorted == false)
              quickSort( data, 0,numData-1,false);
112:        temp = copyData(data, numData);
113:        for (int i = 0; i < numData;i++){
114:              if (temp[i] != MAX_VALUE) {
115:                    tempfrequency[n_freq]=1;
116:                    dummy[n_freq] = temp[i];
117:                    int j = i;
118:                    while ( (j < numData-1) &&
           (temp[j+1] == temp[i])){
119:                          tempfrequency[n_freq] +=1;
120:                                temp[j+1] = MAX_VALUE;
121:                                j = j+1;
122:                          }
123:
124:              n_freq +=1;
125:   }
126:
127:        }
128:        mData= new double[n_freq];
129:        mData = copyData ( dummy, n_freq);
130:        frequency = new int[n_freq];
131:        frequency = copyData(tempfrequency, n_freq);
132:        isClassified = true;
133:        numDataAtLast = n_freq;
```

```
134:  }
135:  /** finding cumulative frequency */
136:  void findCumulativeFrequency(){
137:        // if data are not sorted, sort them decendingly.
138:  if (isSorted == false)
            quickSort( data, 0,numData-1,true);
139:        cFrequency = new int[numDataAtLast];
140:        cFrequency[ numDataAtLast-1] =frequency[numDataAtLast-1];
141:              for(int i = numDataAtLast -2; i >=0; i--){
142:              cFrequency[i] = frequency[i] + cFrequency[i+1];
143:              }
144:              havingCumulatedFreq=true;
145:        }
146:   /** quick sort: sort  all data in ascending or decending
147:   * param item  data to be sorted
148:   * param left  the first item of data
149:   * param right the last item of data
150:   * param ascending  if true , data will be sort in ascending
            othrewise decending
151:   */
152:   void quickSort(double [ ]  item, int left, int right,
            boolean ascending) {
153:         int i,j;
154:         double comparand, temp;
155:                     i = left;
156:                     j = right;
157:                     comparand = item[(left+right)/2];
158:                     do { if (ascending) {
159:                     while( item[i] < comparand && i < right) i++;
160:                     while (comparand < item[j] && j > left) j--;
161:                     }else {
162:                     while( item[i] > comparand && i < right) i++;
163:                     while (comparand > item[j] && j > left) j--;
164:                     }
165:                     if ( i <= j) {
166:                                 temp = item[i];
167:                                 item[i] = item[j];
168:                                 item[j] = temp;
169:                                 i++;
170:                                 j--;
171:                     }
172:                     }while ( i <= j );
173:              if (left < j) quickSort(item, left, j, ascending);
174:              if(i < right ) quickSort(item, i,right, ascending);
175:                           isSorted = true;
176:              }
177:        // ============ Central tendecy =============
178:  /** find mean or average of data */
179:  private void findMean( ) {
180:        double sumData=0;
181:        for(int i = 0 ; i < numData; i++ )
182:              sumData +=data[i];
183:
184:        mean = sumData/(double)numData;
185:  }
186:  /** find median  of data */
187:  private void findMedian( ) {
188:        if (isSorted == false)
                  quickSort(data, 0,numData-1,true);
189:        if (numData%2 == 0)
190:              // number of data is even, find the average
```

```
191:        median = (data[numData/2] + data[(numData/2)-1])/2.0 ;
192:        else
193:              median = data[numData/2];
194:        }
195:
196:  /** find mode  of data */
197:  private void findMode( ) {
198:        int multiMode =1;
199:
200:  if (isClassified == false)                findFrequency( );
201:        if(numData == numDataAtLast)  {
202:              modeData = new double[1];
203:              modeData[0] =0;
204:              mode = 0;
205:              return;
206:        }
207:  // walk through classified data looking for
            double mode or multimode
208:              for (int i = 0; i < numDataAtLast; i++) {
209:              if(frequency[i] >= mode) {
210:                    if (frequency[i] == mode)
211:                          multiMode  +=1;
212:                    else {
213:                    mode = frequency[i];
214:                    multiMode = 1;
215:                    }
216:              }
217:        }
218:        if (multiMode ==1 ) {
219:              modeData = new double[1];
220:        } else {
221:                    modeData = new double[multiMode];
222:      }
223:        int j =0;
224:        for (int i = 0 ; i < numDataAtLast ; i++){
225:                    if (frequency[i] == mode) {
226:                    modeData[j] = mData[i];
227:                    j = j+1;
228:              }
229:        }
230:        }
231:
232:
233:        // ===========Measure of variation =====
234:        /** find range of data */
235:        private void findRange(){
236:              if (isClassified == false)    findFrequency( );
237:              range = data[0] - data[data.length-1];
238:        }
239:
240:   /** find standard deviation of data */
241:   private void findSD() {
242:         double sumDifference=0;
243:         for(int i = 0; i < numData; i++){
244:               sumDifference += (data[i] - mean)*(data[i]-mean);
245:         }
246:         sd = sqrt(sumDifference/(double)(numData-1));
247:   }
248:
249:
250:  // ========== Some parameter inference statistics ==========
```

```
251:
252:        /** Compute the area between 2 statistic z value
253:        * @param lower_z lower limit of area
254:        * @param upper_z upper limit of area
255:        * @ return area under normal curve between 2 value of z
256:        */
257:
258:  public double findAreaUnderNormalCurve(double lower_z,
                              double upper_z) {
259:
260:        Function normalFunction = new Function() {
261:  public double Of(double x) {
262:        return (1/sqrt(2.0*PI)*exp(-(x*x)/2.0));}
263:  };
264:        Simpson1_3Integration si = new Simpson1_3Integration
                (normalFunction,lower_z, upper_z,1000);
265:              return (si.getIntegrationResult());
266:  }
267:
268:  /** finding statistic Z of any data x
269:  * @param x   any data x
270:  * @return z value statistic Z
271:  */
272:  public double findZ(double x){
273:        return ( x - mean)/sd;
274:  }
275:
276:  /** finding statistic Z value at known area under normal curve
277:  * @param area  the area under normal curve should be between 0
to 1.
278:  * @return z value at known area
279:  */
280:  public double findZAtKnownArea (double area)
               throws SimpleStatException {
281:        boolean isAreaGreaterThanHalf = false;
282:        double lowerLimit=0;
283:        double newArea=0;
284:        double deltaX;
285:        double z=0 , sumArea=0;
286:        double dA;
287:              if ( area < 0 || area >1 ) throw
          new SimpleStatException(SimpleStatException.BAD_AREA);
288:              if (area > 0.5){
289:                    newArea = area - 0.5;
290:                    isAreaGreaterThanHalf = true;
291:               } else {
292:                    newArea = 0.5 - area;
293:              }
294:              /*
295:              if (newArea >= 0.3412944) {
296:                   lowerLimit = 1;
297:                   sumArea = 0.3412944;
298:              }
299:              if (newArea >= 0.4771599) {
300:                   lowerLimit = 2;
301:                   sumArea = 0.4771599;
302:              }
303:              if (newArea >= 0.4986253) {
304:                   lowerLimit = 3;
305:                   sumArea = 0.4986253;
306:              }
```

```
307:              */
308:        if (newArea >= 0.01993880583837) {
309:                    lowerLimit = 0.05;
310:                    sumArea = 0.01993880583837;
311:              }
312:              if (newArea >= 0.09870632568292) {
313:                    lowerLimit = 0.25;
314:                    sumArea = 0.09870632568292;
315:              }
316:              if (newArea >= 0.19146246127401) {
317:                    lowerLimit = 0.5;
318:                    sumArea = 0.19146246127401;
319:              }
320:              if (newArea >= 0.34134474606854 ) {
321:                    lowerLimit = 1.0;
322:                    sumArea = 0.34134474606854;
323:              }
324:              if (newArea >= 0.43319279873114) {
325:                    lowerLimit = 1.5;
326:                    sumArea = 0.43319279873114;
327:              }
328:        if (newArea >= 0.47724986805182) {
329:                    lowerLimit = 2;
330:                    sumArea = 0.47724986805182;
331:              }
332:              if (newArea >= 0.49379033467422) {
333:                    lowerLimit = 2.5;
334:                    sumArea = 0.49379033467422;
335:              }
336:              if (newArea >= 0.49865010196837) {
337:                    lowerLimit = 3;
338:                    sumArea = 0.49865010196837;
339:
340:              }
341:              z = lowerLimit;
342:              deltaX = 0.00001;
343:              while( newArea - sumArea >DEFAULT_TOLERANCE) {
344:                    dA = (1/sqrt(2*PI))*0.5*deltaX*(exp(-0.5*z*z)
                      + exp(-0.5*(z+deltaX)*(z+deltaX)));
345:                    sumArea += dA;
346:                    z += deltaX;
347:              }
348:        //    System.out.println(" z in method = " + z);
349:        //    System.out.println(" Sumarea = " + sumArea);
350:
351:              if (isAreaGreaterThanHalf)  return z+deltaX;
352:              else return -(z+deltaX);
353:        }
354:
355:        // ========= Getter ==============
356:  public double getMean(){ return mean;}
357:  public double getMedian() { return median;}
358:  public double getStandardDeviation( ){ return sd;}
359:  public double getRange() { return range;}
360:  public int getFrequencyOfMode() { return mode;}
361:  public double[] getModeData(){ return modeData;}
362:  public double[] getSortedData( ) { return data;}
363:  public double[]getManagedData() { return mData;}
364:  public int[ ] getFrequency() { return frequency;}
365:  public int[] getCumulativeFrequency() { return cFrequency;};
366:  public int getnumDataAtLast() { return numDataAtLast;};
```

```
367:
368:  }
369:
370:
```

เมธอดที่สำคัญของคลาส SimpleStat ได้แก่

findFrequency ( ) ใช้หาความถี่ของข้อมูล ตั้งแต่บรรทัดที่ 104 ถึง บรรทัดที่ 126

findCumulativeFrequency ( ) ใช้หาความถี่สะสมของข้อมูล ตั้งแต่บรรทัดที่ 136 ถึง บรรทัดที่ 145

quickSort(double [ ] item, int left, int right, boolean ascending) ใช้เรียงลำดับข้อมูล จะเรียงลำดับจากค่าน้อยไปค่ามาก ถ้าค่า ascending เป็นจริง และจะเรียงลำดับจากค่ามากไปค่าน้อย ถ้าค่า ascending เป็นเท็จ เมธอดนี้เริ่มตั้งแต่บรรทัดที่ 152 ถึง บรรทัดที่ 176

findMean ( ) เป็นการหาค่าเฉลี่ยของข้อมูล ตั้งแต่บรรทัดที่ 179 ถึง บรรทัดที่ 185

findSD ( ) เป็นการหาค่าส่วนเบี่ยงเบนมาตรฐานของข้อมูล ตั้งแต่บรรทัดที่ 241 ถึง บรรทัดที่ 247

findAreaUnderNormalCurve (double lower_z, double upper_z) เป็นเมธอดที่ใช้หาพื้นที่ใต้เส้นโค้งปกติ ณ ที่ตำแหน่งระหว่างค่า Z 2 ค่า ตั้งแต่บรรทัดที่ 258 ถึง บรรทัดที่ 279

findZAtKnownArea (double area) เป็นเมธอดที่คำนวณหาค่า Z เมื่อทราบพื้นที่ใต้เส้นโค้งปกติ ตั้งแต่บรรทัดที่ 280 ถึง บรรทัดที่ 318

ข้อมูลที่ถูกจัดเรียงลำดับ และหาความถี่เรียบร้อยแล้ว จะเก็บไว้ในตัวแปรอะเรย์ที่ชื่อว่า mData

เพื่อป้องกันข้อผิดพลาดระหว่างการประมวลผลโปรแกรม เช่น ผู้ใช้ให้ค่าพื้นที่ใต้เส้นโค้งปกติมีค่าเกิน 1 หรือ ข้อมูลนี้ไม่สามารถหาค่า mode ได้ เป็นต้น ผู้พัฒนาจึงได้สร้างคลาสรองรับการเกิด exception ของการคำนวณค่าทางสถิติเหล่านี้ ให้ชื่อคลาสเป็น SimpleStatException

Listing 4.4 แสดงรายละเอียดของคลาส SimpleStatException

```
 1:  /*file : SimpleStatException.java
 2:  *  Math Tools in Java
 3:  * by Wachara R.
 4:  * First Released: Mon 07 May 2007
 5:  * Last Updated :
 6:  */
 7:  package MathTools.SimpleStat;
 8:  public class SimpleStatException extends Exception {
 9:        public static final String NO_MODE
                 = "THIS BUNDLE OF DATA HAS NO MODE";
10:        public static final String BAD_AREA
           = " AREA UNDER NORMAL CURVE MUST BE BETWEEN 0 to 1.";
11:
12:        /**
```

```
13:         * constructor.
14:         * @param err the error message
15:         */
16:         public SimpleStatException (String err ){
17:               super(err);
18:         }
19:
20:   }
```

เนื่องจากการคำนวณคะแนน T ปกติต้องเกี่ยวข้องกับการคำนวณหาพื้นที่ใต้เส้นโค้งปกติจึงได้ทำการทดสอบค่าพื้นที่ใต้เส้นโค้ง เมื่อ Z มีค่าต่าง ๆ กัน แล้วนำไปเปรียบเทียบกับตารางความน่าจะเป็นสะสมของการแจกแจงปกติมาตรฐาน(Z) ในหนังสือสถิติว่าได้ค่าต่างกันหรือไม่อย่างไร (คิดที่ทศนิยม 4 ตำแหน่ง)

เพื่อไม่ให้ล้นหน้ากระดาษ จะแสดงผลการคำนวณหาค่า Z หาพื้นที่ใต้โค้งเป็น 2 ตาราง คือตารางที่ Z มีค่าเป็นบวก (ตั้งแต่ 0 ถึง 1.99) และและตารางพื้นที่ใต้โค้งที่ Z มีค่าเป็นลบ (ตั้งแต่ -1.99 ถึง 0) โดยอาศัยโปรแกรม Ztable.java ซึ่งมีรายละเอียดของโปรแกรมดังนี้

Listing 4.5 แสดงรายละเอียดของคลาส Ztable

```
 1:  /* File :Ztable.java
 2:  * Project: Math Tools in Java
 3:  * Purpose: Test class for finding Z table
 4:  * Author : Wachara R.
 5:  * First Released: Th 10 May 2007
 6:  * Last Updated :
 7:  */
 8:
 9:  import static java.lang.Math.*;
10:  import java.io.*;
11:  import java.util.*;
12:  import MathTools.Common.*;
13:  import MathTools.SimpleStat.SimpleStat;
14:
15:  public class Ztable {
16:
17:  public static void main(String[] args) {
18:  double z,area, step;
19:  double tempz;
20:        SimpleStat s = new SimpleStat();
21:              step = 0.01;
22:        System.out.println("\n\t  Find the Area under Normal
          curve  between 0 to 1.99 step 0.01.\n\n");
23:        System.out.println("         0      1      2      3
    4      5      6      7      8      9");
24:        i = 0;
25:        area = 0;
26:        for( z = 0; z<= 2; z = z+step) {
27:              if ( i % 10 ==0)  {
28:                    System.out.printf("\n%2.1f  ",z);
29:                     i =0;
30:              }
31:              i +=1;
```

```
32:               area = s.findAreaUnderNormalCurve(0,z);
33:               System.out.printf(" %4.4f",area+0.5);
34:         }
35:         System.out.println("\n\t  Find the Area under Normal
            curve between 0 to -1.99 step 0.01.\n\n");
36:         System.out.println("         0      1      2      3
    4       5      6      7      8      9");
37:         area = 0;
38:         step = 0.1;
39:         for( z = -1.9; z <= 0; z = z+step) {
40:               System.out.printf("\n%2.1f  ",z);
41:                tempz = z;
42:               for(int k = 0; k < 10 ; k++) {
43:                     area = s.findAreaUnderNormalCurve(z,0);
44:                     System.out.printf(" %4.4f", 0.5 - area);
45:                     tempz = tempz + 0.01;
46:               }
47:
48:         }
49:
50:   }
51:   }
```

ผลการทำงานของโปรแกรมจะเป็นดังนี้ ตารางแสดงพื้นที่ใต้เส้นโค้งปกติของค่า Z ตั้งแต่ 0 ถึง 1.99 เพิ่มค่าครั้งละ 0.01

	0	1	2	3	4	5	6	7	8	9
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767

ตารางต่อไปนี้แสดงพื้นที่ใต้เส้นโค้งปกติ ตั้งแต่ -1.99 ถึง 0

	0	1	2	3	4	5	6	7	8	9
-1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0239	0.0233
-1.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0301	0.0294
-1.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
-1.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
-1.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0571	0.0559
-1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681
-1.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0838	0.0823
-1.2	0.1151	0.1131	0.1112	0.1093	0.1075	0.1056	0.1038	0.1020	0.1003	0.0985
-1.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
-1.0	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379
-0.9	0.1841	0.1814	0.1788	0.1762	0.1736	0.1711	0.1685	0.1660	0.1635	0.1611
-0.8	0.2119	0.2090	0.2061	0.2033	0.2005	0.1977	0.1949	0.1922	0.1894	0.1867
-0.7	0.2420	0.2389	0.2358	0.2327	0.2296	0.2266	0.2236	0.2206	0.2177	0.2148
-0.6	0.2743	0.2709	0.2676	0.2643	0.2611	0.2578	0.2546	0.2514	0.2483	0.2451
-0.5	0.3085	0.3050	0.3015	0.2981	0.2946	0.2912	0.2877	0.2843	0.2810	0.2776
-0.4	0.3446	0.3409	0.3372	0.3336	0.3300	0.3264	0.3228	0.3192	0.3156	0.3121
-0.3	0.3821	0.3783	0.3745	0.3707	0.3669	0.3632	0.3594	0.3557	0.3520	0.3483
-0.2	0.4207	0.4168	0.4129	0.4090	0.4052	0.4013	0.3974	0.3936	0.3897	0.3859
-0.1	0.4602	0.4562	0.4522	0.4483	0.4443	0.4404	0.4364	0.4325	0.4286	0.4247

เมื่อเปรียบเทียบกับตารางหาค่าสถิติ Z จากหนังสือ การวิจัยและวิเคราะห์ข้อมูลทางสถิติด้วย SPSS. ของ ผศ. ธานินทร์ ศิลป์จารุ (3:220). พบว่าได้ตรงกันทุกค่า (คิดที่ทศนิยม 4 ตำแหน่ง)

เปรียบเทียบค่าที่คำนวณได้กับผลลัพธ์จากการใช้ฟังก์ชัน NormSDist คำนวณพบว่ามีค่าตรงกันทุกค่า (คิดที่ทศนิยม 4 ตำแหน่ง) ดูภาคผนวก A หน้า 88

ในทางกลับกัน ถ้าทราบพื้นที่ใต้เส้นโค้งปกติ ต้องการทราบค่า Z ตรงตำแหน่งที่ทำให้เกิดพื้นที่ใต้เส้นโค้งค่านี้ สามารถใช้ฟังก์ชัน findZAtKnownArea( ) หาย้อนกลับได้เช่นเดียวกัน ทดสอบค่าที่คำนวณได้จากโปรแกรมเทียบกับผลลัพธ์ที่ได้จากการใช้ฟังก์ชัน NormSInv พบว่ามีค่าตรงกันทุกค่า โดยคิดค่า Z ที่ทศนิยม 2 ตำแหน่ง ดูภาคผนวกที่ A หน้า 95

4.3 **ส่วนที่คำนวณค่าคะแนนทีปกติ และเลือกพิสัยของระดับคะแนน**

คลาสที่ใช้คำนวณคะแนนทีปกติ และกำหนดระดับคะแนนคือ คลาส TScore โดยพัฒนาคลาสนี้สืบทอดมาจากคลาส SImpleStat เมธอดที่เพิ่มขึ้นมาคือ findPercentile() ใช้คำนวณหาค่าเปอร์เซ็นต์ไทล์ ตั้งแต่บรรทัดที่ 51 ถึง บรรทัดที่ 65 computeTScore() คำนวณหาคะแนนทีปกติ ตั้งแต่บรรทัดที่ 66 ถึง บรรทัดที่ 80 assessGrade( ) และกำหนดระดับคะแนนตามช่วงที่ผู้ใช้กำหนด ตั้งแต่บรรทัดที่ 81 ถึง บรรทัดที่ 160

Listing 4.6 แสดงรายละเอียดของคลาส Tscore

```
 1:  /* File : TScore.java
 2:  * Project: Math Tools in Java
 3:  * Purpose: Compute Normalized Tscore and
 4:  *                 and Grade Assesment
 5:  * Author : Wachara R.
 6:  * First Released: Sun 9 March 2008
 7:  * Last Updated : Tue 24 May 2008
 8:  */
 9:
10:  package MathTools.SimpleStat;
11:
12:  import static java.lang.Math.*;
13:  import static java.lang.Double.*;
14:  import MathTools.SimpleStat.SimpleStat;
15:  import MathTools.SimpleStat.SimpleStatException;
16:
17:  /** class for computing nomalized Tscore and
18:  *       appropriated grade  assesment
19:  *     */
20:
21:  public  class TScore extends SimpleStat {
22:  /** Percentile */ private double [ ] Percentile;
23:  /** TScore */      private double [ ] Tscore;
24:  /** Grade Option */  private int gradeOption=0;
25:  /** Grade assessment of each score */ private String [ ] grade;
26:
27:  /** Constructor
28:  * @param dat data for computing basic statistics
29:  */
30:  /** Constructor */
31:  public TScore( ) {
32:  super();
33:  }
34:
35:  public TScore( double[ ] dat) {
36:  super(dat);
37:  findPercentile();
38:  computeTScore();
39:  assessGrade();
40:  }
41:
42:  public TScore( double[ ] dat, int nData, int grOption) {
43:  super(dat,nData);
44:  gradeOption = grOption;
45:  findPercentile();
46:  computeTScore();
47:  assessGrade();
48:  }
49:  /** finding  percentiles*/
50:
51:  void findPercentile(){
52:  // if data are not sorted, sort them descendingly.
53:  if (isSorted == false)  quickSort( data, 0,numData-1,true);
54:  // if data  are not computed cumulated frequency , compute them
55:  if(havingCumulatedFreq=false) findCumulativeFrequency();
56:  Percentile = new double[numData];
57:
58:  for(int i = numDataAtLast -1; i >=0; i--){
59:  if ( i == numDataAtLast-1)
```

```
  Percentile[i] = 0.5*frequency[i]/numData ;
else
Percentile[i] = (cFrequency[i+1] + 0.5* frequency[i] )/numData;
}
}

void computeTScore() {
Tscore = new double[numDataAtLast];
try
{
      for (int i=0;i < numDataAtLast;i++ )            {
            Tscore[i]=  findZAtKnownArea (Percentile[i]) ;
            // Change  z value to T score.
            Tscore[i] = Tscore[i]*10.0 +50.0;
      }
}
catch (SimpleStatException err)
{     }

}

void  assessGrade( )  {
double range;

    range = (Tscore[0] - Tscore[numDataAtLast -1]) / 5.0;
    if (range < 0)   range = -range;
   grade = new String[numDataAtLast];
    switch ( gradeOption)    {
    case 0 : {
      for(int i = 0 ; i < numDataAtLast; i++) {
            grade[ i ] = "F";
            if (Tscore[i] >= 50.0 -1.5*range)  grade[i] = "D";
            if (Tscore[i] >= 50.0 -1.0*range)  grade[i] = "D+";
            if (Tscore[i] >= 50.0 -0.5*range)  grade[i] = "C";
            if (Tscore[i] >= 50.0)  grade[i] = "C+";
            if (Tscore[i] >= 50.0 +0.5*range)  grade[i] = "B";
            if (Tscore[i] >= 50.0 +1.0*range)  grade[i] = "B+";
            if (Tscore[i] >= 50.0 +1.5*range)  grade[i] = "A";
      }
      break;
      }
     case 1: {
      for(int i = 0 ; i < numDataAtLast; i++) {
            grade[ i ] = "F";
            if (Tscore[i] >= 50.0 -1.5*range)  grade[i] = "D";
            if (Tscore[i] >= 50.0 -0.5*range)  grade[i] = "C";
            if (Tscore[i] >= 50.0 +0.5*range)  grade[i] = "B";
            if (Tscore[i] >= 50.0 +1.5*range)  grade[i] = "A";
      }
      break;
      }
    case 2: {
      for(int i = 0 ; i < numDataAtLast; i++) {
            grade[ i ] = "D";
            if (Tscore[i] >= 50.0 -1.5*range)  grade[i] = "D+";
            if (Tscore[i] >= 50.0 -1.0*range)  grade[i] = "C";
            if (Tscore[i] >= 50.0 -0.5*range)  grade[i] = "C+";
            if (Tscore[i] >= 50.0)  grade[i] = "B";
            if (Tscore[i] >= 50.0 +0.5*range)  grade[i] = "B+";
            if (Tscore[i] >= 50.0 +1.0*range)  grade[i] = "A";
      }
```

```
121:        break;
122:        }
123:
124:      case 3: {
125:        for(int i = 0 ; i < numDataAtLast; i++) {
126:              grade[ i ] = "D";
127:              if (Tscore[i] >= 50.0 -1.0*range)  grade[i] = "C";
128:              if (Tscore[i] >= 50.0)  grade[i] = "B";
129:              if (Tscore[i] >= 50.0 +1.0*range)  grade[i] = "A";
130:        }
131:        break;
132:        }
133:        case 4: {
134:        for(int i = 0 ; i < numDataAtLast; i++) {
135:              grade[ i ] = "C";
136:              if (Tscore[i] >= 50.0 -1.0*range)  grade[i] = "C+";
137:              if (Tscore[i] >= 50.0 -0.5*range)  grade[i] = "B";
138:              if (Tscore[i] >= 50.0)  grade[i] = "B+";
139:              if (Tscore[i] >= 50.0 +0.5*range)  grade[i] = "A";
140:        }
141:        break;
142:        }
143:   case 5 : {
144:        for(int i = 0 ; i < numDataAtLast; i++) {
145:              grade[ i ] = "C";
146:              if (Tscore[i] >= 50.0 -0.5*range)  grade[i] = "B";
147:              if (Tscore[i] >= 50.0 +0.5*range)  grade[i] = "A";
148:        }
149:        break;
150:        }
151:   case 6 : {
152:        for(int i = 0 ; i < numDataAtLast; i++) {
153:              grade[ i ] = "B";
154:              if (Tscore[i] >= 50.0 )  grade[i] = "A";
155:        }
156:        break;
157:        }
158:  } //  end switch
159:  }
160:
161:  public double[ ] getPercentile( ) { return Percentile;}
162:  public double[ ] getTScore( ) { return Tscore;}
163:  public String[ ] getGrade() {return grade; }
164:  public int getGradeOption() { return gradeOption;}
165:  public void setGradeOption(int gOption) {
166:  gradeOption = gOption;
167:        if(gradeOption < 0 || gradeOption > 6)
168:              gradeOption=0;
169:  }
170:
171:  }
```

การกำหนดระดับคะแนนหรือตัดเกรดนั้น ผู้ใช้สามารถเลือกได้ว่าต้องการจะตัดเกรดกี่เกรด ในตัวโปรแกรมจะเก็บค่าที่ผู้ใช้เลือกไว้ในตัวแปรชื่อ gradeOption โดยปกติจะตั้งไว้ให้ตัดเกรดเป็น A, B+, B, C+, C, D+, D, F

หลักการตัดเกรดจะยึดคะแนนทีปกติ ที่ 50 เป็นหลักเสมอ แล้วแบ่งพื้นที่ใต้เส้นโค้งปกติให้เป็นส่วน ๆ แล้วกำหนดเกรดให้แก่แต่ละส่วนนั้น

เริ่มต้นด้วยการกำหนดพิสัยของเกรด (range) โดยคำนวณค่าพิสัย ในบรรทัดที่ 84

$$\text{พิสัย} = \frac{\text{คะแนนทีปกติที่มีค่าสูงสุด} - \text{คะแนนทีปกติที่มีค่าต่ำสุด}}{\text{จำนวนพื้นที่ใต้โค้งที่แบ่งเพื่อตัดเกรด}}$$

เมื่อผู้ใช้เลือกตัดเกรด A, B+, B, C+, C, D+, D, F (ซึ่งเป็นค่า default)

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 + 1.5 * พิสัย จะได้เกรด A

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 + 1.0* พิสัย แต่ไม่ถึง 50 +1.5* พิสัย จะได้เกรด B+

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 + 0.5* พิสัย แต่ไม่ถึง 50 +1.0* พิสัย จะได้เกรด B

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 แต่ไม่ถึง 50 +0.5* พิสัย จะได้เกรด C+

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 - 0.5*พิสัย แต่ไม่ถึง 50 จะได้เกรด C

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 -1.0*พิสัย แต่ไม่ถึง 50 - 0.5* พิสัย จะได้เกรด D+

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 -1.5*พิสัย แต่ไม่ถึง 50 – 1.0* พิสัย จะได้เกรด D

คะแนนทีปกติ ที่น้อยกว่า 50 -1.5*พิสัย จะได้เกรด F

เมื่อผู้ใช้เลือกตัดเกรด A, B, C, D, F

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 + 1.5 * พิสัย จะได้เกรด A

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 + 0.5* พิสัย แต่ไม่ถึง 50 +1.5* พิสัย จะได้เกรด B

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 - 0.5*พิสัย แต่ไม่ถึง 50 + 0.5*พิสัย จะได้เกรด C

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 -1.5*พิสัย แต่ไม่ถึง 50 – 0.5* พิสัย จะได้เกรด D

คะแนนทีปกติ ที่น้อยกว่า 50 -1.5*พิสัย จะได้เกรด F

เมื่อผู้ใช้เลือกตัดเกรด A, B+, B, C+, C, D+, D ไม่มี F

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 + 1.0 * พิสัย จะได้เกรด A

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 + 0.5* พิสัย แต่ไม่ถึง 50 +1.0* พิสัย จะได้เกรด B+

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 แต่ไม่ถึง 50 +0.5* พิสัย จะได้เกรด B

เมื่อคะแนนทีปกติ เท่ากับหรือมากกว่า 50 - 0.5*พิสัย แต่ไม่ถึง 50 จะได้เกรด C+

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 -1.0*พิสัย แต่ไม่ถึง 50 - 0.5* พิสัย จะได้เกรด C

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 -1.5*พิสัย แต่ไม่ถึง 50 – 1.0* พิสัย จะได้เกรด D+

คะแนนทีปกติ ที่น้อยกว่า 50 -1.5*พิสัย จะได้เกรด D

เมื่อผู้ใช้เลือกตัดเกรด A, B, C, D

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 + 1.0 * พิสัย จะได้เกรด A

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 แต่ไม่ถึง 50 +1.0* พิสัย จะได้เกรด B

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 – 1.0*พิสัย แต่ไม่ถึง 50 จะได้เกรด C

คะแนนทีปกติ ที่น้อยกว่า 50 -1.0*พิสัย จะได้เกรด D

เมื่อผู้ใช้เลือกตัดเกรด A, B+, B, C+, C

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 + 0.5* พิสัย จะได้เกรด A

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 แต่ไม่ถึง 50 +0.5* พิสัย จะได้เกรด B+

เมื่อคะแนนทีปกติ เท่ากับหรือมากกว่า 50 - 0.5*พิสัย แต่ไม่ถึง 50 จะได้เกรด B

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 -1.0*พิสัย แต่ไม่ถึง 50 - 0.5* พิสัย จะได้เกรด C+

คะแนนทีปกติ ที่น้อยกว่า 50 -1.0*พิสัย จะได้เกรด C

เมื่อผู้ใช้เลือกตัดเกรด A, B, C, D

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 + 0.5 * พิสัย จะได้เกรด A

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 – 0.5*พิสัย แต่ไม่ถึง 50 +0.5*พิสัย จะได้เกรด B

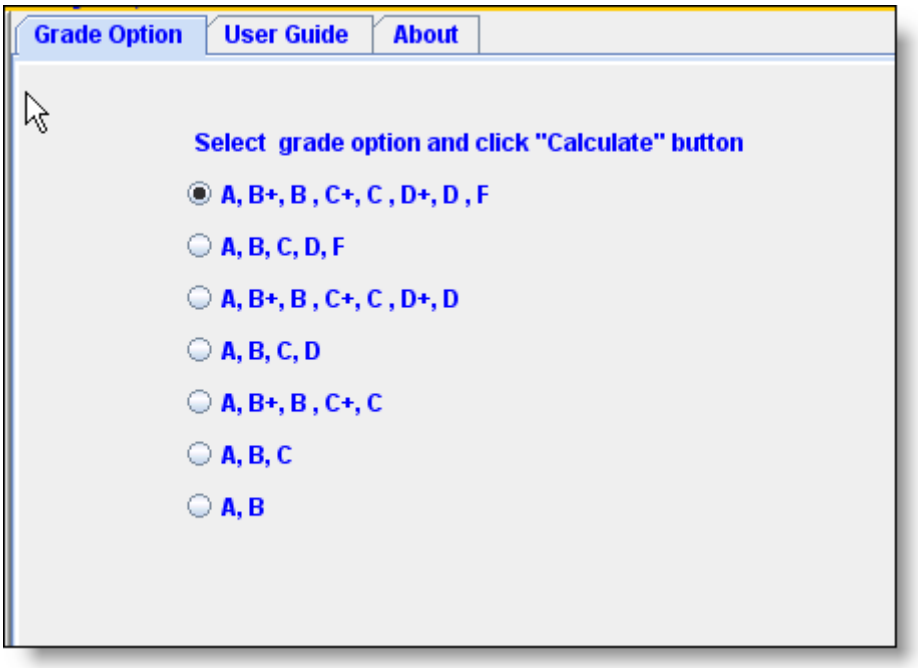
คะแนนทีปกติ ที่น้อยกว่า 50 -0.5*พิสัย จะได้เกรด C

เมื่อผู้ใช้เลือกตัดเกรด A, B

คะแนนทีปกติ เท่ากับหรือมากกว่า 50 จะได้เกรด A

คะแนนทีปกติ ที่น้อยกว่า 50 จะได้เกรด B

การระบุจำนวนเกรดที่จะตัด ผู้ใช้สามารถเลือกได้จากตัวเลือกที่ทำเป็นปุ่ม radio button คลาสที่ทำหน้าที่รับรู้ตัวเลือกที่ผู้ใช้กำหนดคือ คลาส GradeOptionPanel เมื่อแสดงผลบนจอภาพจะเห็นแผงหน้าปัดด้านซ้ายมือ จะมีลักษณะดังรูปที่ 4.2



รูปที่ 4.2 แสดงกลุ่ม radio button ที่ผู้ใช้สามารถเลือกจำนวนเกรดที่จะกำหนดได้

รายละเอียดของคลาส GradeOptionPanel มีดังนี้

Listing 4.7 แสดงรายละเอียดของคลาส GradeOptionPanel

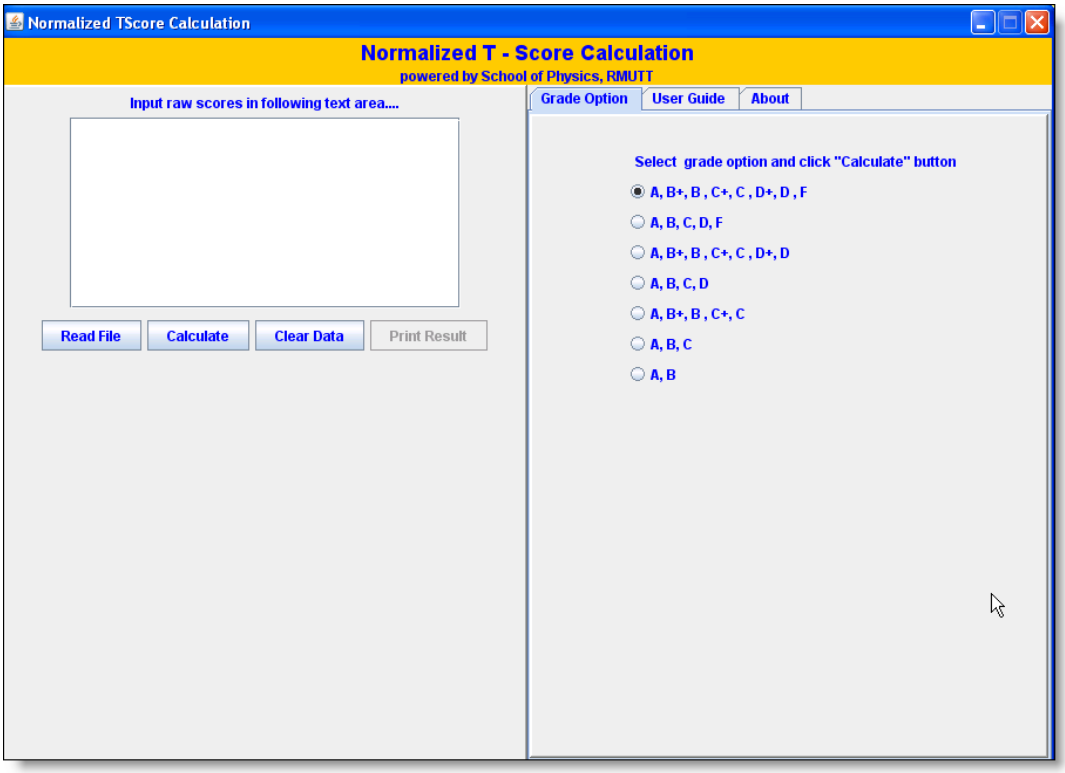
```
 1:  /* File :GradeOptionPanel.java
 2:  * Project: Normalized T Score.
 3:  * Purpose: Select the ranges of grade assessment
 4:  * Author : Wachara R.
 5:  * First Released: Sun 25 May 2008
 6:  * Last Updated :  Sun 25 May 2008
 7:  */
 8:  package JApp.TS;
 9:  import javax.swing.*;
10:  import java.awt.*;
11:  import java.awt.event.*;
12:
13:  public class GradeOptionPanel extends JPanel
                 implements ActionListener {
14:  private  int gradeRangeOption=0;
15:
16:  public GradeOptionPanel () {
17:          initComponents();
18:      }
19:
```

```
20:      private void initComponents() {
21:        subPanel = new JPanel( new GridLayout (15,0) );
22:subPanel.setBorder(BorderFactory.createEmptyBorder(20,20,10,10));
           // top, left , bottom, right
23:        //setBorder(BorderFactory.createEtchedBorder());
24:        setBorder(BorderFactory.createRaisedBevelBorder());
25:
26:        labelPanel = new JPanel();
27:        JLabel selectLabel = new JLabel(" Select  grade
           option and click \"Calculate\" button");
28:
29:
30:        selectLabel.setForeground(Color.blue);
31:  //    selectLabel2.setForeground(Color.blue);
32:        gradeOption0 = new   JRadioButton ("A, B+, B , C+, C ,
                  D+, D , F     ",true);
33:        gradeOption0.setForeground(Color.blue);
34:         gradeOption1= new   JRadioButton ("A, B, C, D, F");
35:         gradeOption1.setForeground(Color.blue);
36:        gradeOption2= new   JRadioButton ("A, B+, B , C+, C ,
                  D+, D");
37:         gradeOption2.setForeground(Color.blue);
38:        gradeOption3= new   JRadioButton ("A, B, C, D");
39:        gradeOption3.setForeground(Color.blue);
40:        gradeOption4= new   JRadioButton ("A, B+, B , C+, C ");
41:        gradeOption4.setForeground(Color.blue);
42:        gradeOption5= new   JRadioButton ("A, B, C ");
43:        gradeOption5.setForeground(Color.blue);
44:        gradeOption6= new   JRadioButton ("A, B");
45:        gradeOption6.setForeground(Color.blue);
46:        group1 = new ButtonGroup();
47:
48:        group1.add(gradeOption0);
49:        group1.add(gradeOption1);
50:        group1.add(gradeOption2);
51:        group1.add(gradeOption3);
52:        group1.add(gradeOption4);
53:        group1.add(gradeOption5);
54:        group1.add(gradeOption6);
55:
56:        //Register a listener for the radio buttons.
57:          gradeOption0.addActionListener(this);
58:          gradeOption1.addActionListener(this);
59:          gradeOption2.addActionListener(this);
60:          gradeOption3.addActionListener(this);
61:          gradeOption4.addActionListener(this);
62:          gradeOption5.addActionListener(this);
63:          gradeOption6.addActionListener(this);
64:
65:        labelPanel.add(selectLabel);
66:        //labelPanel.add(selectLabel2);
67:        subPanel.add(labelPanel);
68:        subPanel.add(gradeOption0);
69:        subPanel.add(gradeOption1);
70:        subPanel.add(gradeOption2);
71:        subPanel.add(gradeOption3);
72:        subPanel.add(gradeOption4);
73:        subPanel.add(gradeOption5);
74:        subPanel.add(gradeOption6);
75:        this.add(subPanel);
76:
```

```
 77:
 78:  }
 79:
 80:   /** Listens to the radio buttons. */
 81:      public void actionPerformed(ActionEvent e) {
 82:       if (gradeOption0.isSelected())  gradeRangeOption=0;
 83:   if (gradeOption1.isSelected())  gradeRangeOption=1;
 84:   if (gradeOption2.isSelected())  gradeRangeOption=2;
 85:   if (gradeOption3.isSelected())  gradeRangeOption=3;
 86:   if (gradeOption4.isSelected())  gradeRangeOption=4;
 87:   if (gradeOption5.isSelected())  gradeRangeOption=5;
 88:   if (gradeOption6.isSelected())  gradeRangeOption=6;
 89:      }
 90:
 91:  public int getGradeOption() {
 92:        return gradeRangeOption;
 93:  }
 94:  private JPanel subPanel ; // panel for making space at the edge
 95:  private  JPanel labelPanel;
 96:  private  ButtonGroup group1;
 97:  private  JRadioButton gradeOption0;
 98:  private  JRadioButton gradeOption1,gradeOption2,gradeOption3;
 99:  private JRadioButton gradeOption4, gradeOption5, gradeOption6;
100:
101:  }
102:
```

4.4 **ส่วนที่แสดงผลลัพธ์**

เมื่อให้โปรแกรมเริ่มต้นทำงาน โปรแกรมจะแสดงผลบนจอภาพดังรูปที่ 4.3 พื้นที่หน้าต่างของโปรแกรมจะถูกแบ่งเป็น ส่วนหัวของหน้าต่างซึ่งจะแสดงชื่อโปรแกรมด้วยตัวอักษรสีน้ำเงินบนพื้นสีส้ม พื้นที่ใต้ส่วนหัวของหน้าต่างจะถูกแบ่งออกเป็น 2 ส่วน ด้านซ้าย จะเป็นพื้นที่ที่ใช้ป้อนข้อมูลและแสดงผลลัพธ์ที่ได้จากการคำนวณ ส่วนด้านขวามือ จะเป็นแผ่นป้ายที่มีชื่อติดอยู่ที่มุมบนซ้าย หรือเรียกสั้น ๆ ว่า แท็บ (Tabbed Pane) เรียงซ้อนกันอยู่ 3 แผ่น เป็นแผ่นที่ใช้เลือกจำนวนเกรด (Grade option) แผ่นที่บอกวิธีใช้งานโปรแกรม (User guide) และแผ่นที่บอกชื่อผู้พัฒนาโปรแกรม (About)



รูปที่ 4.3 แสดงหน้าตาของโปรแกรมเมื่อเริ่มใช้งาน

เมื่อมองในรายละเอียด การออกแบบหน้าจอ ได้กำหนดให้คลาส TScorePanel เป็นคลาสหลัก จะเป็นคลาสที่รวบรวมเรียกใช้คลาสต่าง ๆ ที่เกี่ยวข้องเข้าไว้ด้วยกัน คลาส TScorePanel เป็นคลาสที่สืบทอดมาจากคลาสแม่ คือ JPanel ซึ่งเป็นคลาสที่มีอยู่แล้วใน API ของจาวา การจัดวาง Layout ของ TScorePanel เป็นแบบ BorderLayout จากนั้นจึงจัดวางแผงหน้าปัด หรือ Panel อื่น ๆ ซ้อนทับ TScorePanel นี้เป็นชั้น ๆ ขึ้นไป ได้แก่

SubMainPanel จัดวาง Layout แบบ GridLayout เป็นการแบ่งหน้าต่างของโปรแกรมเป็น 2 ส่วนคือ แผงหน้าปัดด้านซ้ายสำหรับใส่ leftPanel และแผงหน้าปัดด้านขวาสำหรับ TabbedPane การวาง subMainPanel เป็นการวางลงไปตรงบริเวณที่เป็นพื้นที่ตรงกลางของ TScorePanel (BorderLayout.CENTER)

LeftPanel เป็น Panel ที่อยู่ด้านซ้ายมือของหน้าต่างโปรแกรม แผงหน้าปัดนี้จะเป็น container ที่ใช้บรรจุ หรือวาง Panel ต่าง ๆ ตามลำดับจากด้านบนลงไปด้านล่างดังนี้ ScoreTextArea พร้อมกับ ScrollPane buttonPanel, statPanel และ tablePanel

TabbedPane เป็นแผ่นป้ายหรือ แท็บ จำนวน 3 แผ่น แต่ละแผ่นจะเป็นที่บรรจุของ GradeOptionPanel, helpPanel และ aboutPanel ตามลำดับ

ScoreTextArea พร้อม ScrollPane จะปรากฏให้เห็นเป็นพื้นที่รูปสี่เหลี่ยมพื้นหลังเป็นสีขาวบนจอภาพ บริเวณนี้จะเป็นพื้นที่รับข้อมูลที่ผู้ใช้พิมพ์เข้ามา หรือได้มาจากการอ่านไฟล์ที่เก็บข้อมูลไว้ในฮาร์ดดิสก์ หรือใน Data storage ใด ๆ ถ้าจำนวนข้อมูลมีจนล้นพื้นที่ที่กำหนดไว้ จะปรากฏ ScrollPane ให้สามารถใช้เมาส์เลื่อนขึ้นลง เพื่อดูข้อมูลส่วนที่ล้นกรอบของ Text area ได้

ButtonPanel ใช้สำหรับวางปุ่มคำสั่งต่าง ๆ ได้แก่

ปุ่ม Read file เป็นปุ่มที่ทำให้โปรแกรมไปเรียกและอ่านไฟล์ข้อมูล นำข้อมูลมาแสดงใน Text area

ปุ่ม Calculate เมื่อกดปุ่มนี้ โปรแกรมจะเริ่มทำการตรวจสอบข้อมูลคะแนนดิบ ว่าถูกต้องตามกฎเกณฑ์ที่กำหนดไว้หรือไม่ ถ้าข้อมูลไม่มีข้อผิดพลาด โปรแกรมจะประมวลผลโดยเริ่มจากการจัดเรียงลำดับข้อมูล หาความถี่ ความถี่สะสม เปอร์เซนต์ไทล์ และ คะแนน T ปกติ พร้อมกับกำหนดระดับคะแนนหรือเกรดให้แก่ข้อมูลแต่ละค่า เมื่อคำนวณเสร็จจะแสดงผลลัพธ์ที่คำนวณเป็นตารางตรงบริเวณใต้ buttonPanel

ปุ่ม Clear Data ใช้เมื่อต้องการลบข้อมูลใน Text area หรือต้องการอ่านไฟล์ข้อมูลอันใหม่

ปุ่ม Print result เป็นการพิมพ์ผลลัพธ์ที่แสดงอยู่ในตารางออกทางเครื่องพิมพ์ ตอนเริ่มต้นโปรแกรม ปุ่มนี้จะมีลักษณะเป็นสีจาง ๆ ซึ่งไม่สามารถใช้เมาส์คลิกให้ทำงานได้ จนกว่าผู้ใช้จะกดปุ่ม Calculate ปุ่มนี้จึงจะสามารถทำงานได้ (enabled)

StatPanel แผงหน้าปัดนี้จะอยู่ใต้ buttonPanel มีไว้สำหรับแสดงค่าทางสถิติ ได้แก่ ค่าเฉลี่ย และส่วนเบี่ยงเบนมาตรฐานของคะแนนดิบ

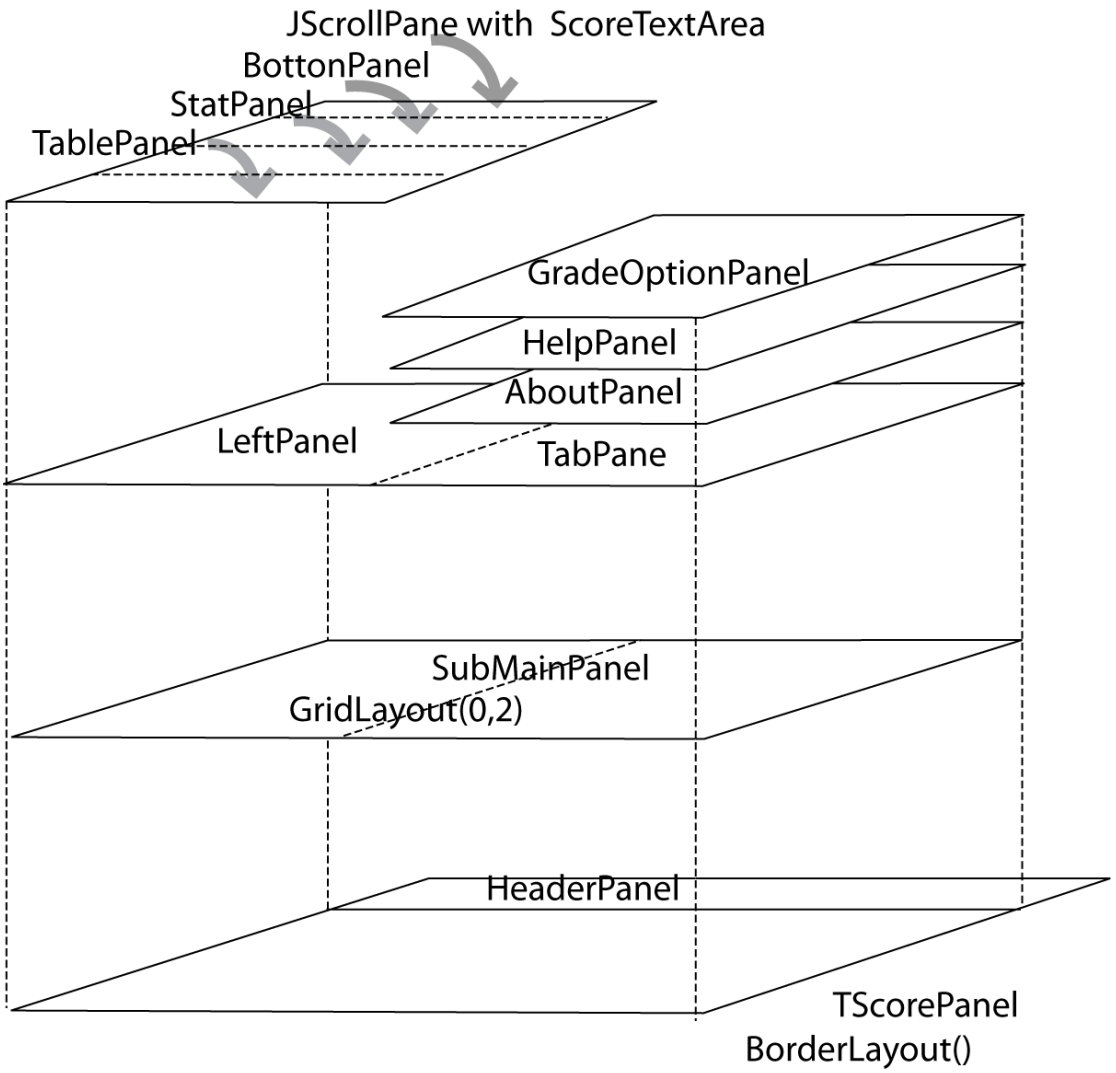
TablePanel ใช้สำหรับเป็นที่วางตารางแสดงผลลัพธ์ที่ได้จากการคำนวณ ถ้าข้อมูลที่ปรากฏบนตารางมีจำนวนแถวมากกว่าพื้นที่ของตาราง ก็จะมี Scrollbar สำหรับเลื่อนขึ้นลงเพื่อดูผลลัพธ์ตรงส่วนที่ล้นออกไปนอกพื้นที่ตารางได้

GradeOptionPanel เป็น Panel แผ่นแรกที่อยู่ทางด้านขวามือ เป็นแผงหน้าปัดที่ใช้แสดงปุ่ม radio button บอกถึงจำนวนเกรดที่ผู้ใช้ต้องการตัด

HelpPanel เป็น Panel ที่แสดงขั้นตอนการใช้งานโปรแกรมนี้

AboutPanel เป็น Panel ที่บอกเวอร์ชันของโปรแกรม

การวางแผงหน้าปัด หรือPanel ต่าง ๆ สามารถเขียนเป็นภาพได้ดังนี้



รูปที่ 4.4 การวาง Panel ซ้อนทับลงไปใน TScorePanel

Listing 4.8 แสดงรายละเอียดคลาส TScorePanel ซึ่งเป็น container หลักของ panel อื่น ๆ

```
 1:  /* File :TScorePanel.java
 2:  * Project: Math Tools in Java
 3:  * Purpose: Test class for finding TScore in applet mode
 4:  * Author : Wachara R.
 5:  * First Released: Sat 17 May 2008
 6:  * Last Updated :  Fri 13 June 2008
 7:  */
 8:  package JApp.TS;
 9:  import static java.lang.Math.*;
10:  import MathTools.SimpleStat.SimpleStat;
11:  import MathTools.SimpleStat.TScore;
12:  import MathTools.SimpleStat.ScoreTextArea;
13:  import static MathTools.SimpleStat.TScoreConstants.*;
14:  import javax.swing.*;
15:  import javax.swing.table.*;
```

```
16:  import java.awt.*;
17:  import java.awt.event.*;
18:  import java.awt.print.*;
19:  import java.util.*;  // for date and time
20:  import java.io.*; // for reading data from a file.
21:  public class TScorePanel extends JPanel {
22:
23:  public TScorePanel() {
24:          initComponents();
25:      }
26:
27:      private void initComponents() {
28:        /* set header label at the top of the frame
29:        *  and put them on header Panel
30:        */
31:
32:  headerLabel = new JLabel("Normalized T - Score
     Calculation",JLabel.CENTER);
33:  headerLabel2=new JLabel("powered by School of Physics, RMUTT",
     JLabel.CENTER);
34:  headerLabel.setFont(new Font("Dialog", Font.BOLD, 18));
35:  headerLabel2.setFont(new Font("Dialog", Font.BOLD, 12));
36:  headerLabel.setForeground(Color.blue);
37:  headerLabel2.setForeground(Color.blue);
38:
39:        headerPanel = new JPanel(new BorderLayout());
40:        headerPanel.add(headerLabel, BorderLayout.CENTER);
41:        headerPanel.add(headerLabel2, BorderLayout.SOUTH);
42:        headerPanel.setBackground(Color.orange);
43:
44:  /* Create leftPanel which consists of  inputLabel ,
      scoreTextArea, buttonPanel
45:  * statPanel and  tablePanel
46:  */
47:  JLabel inputLabel = new JLabel ("Input raw scores in following
     text area....");
48:  inputLabel.setForeground(Color.blue);
49:        /* ScoreTextArea */
50:        scoreText = new  ScoreTextArea();
51:        scoreText.setColumns(30);
52:        scoreText.setRows(10);
53:        jScrollPane1 = new  JScrollPane(scoreText);
54:
55:  /* buttons  and buttonPanel*/
56:        cmdReadFile = new JButton("Read File");
57:        cmdReadFile.setForeground(Color.blue);
58:
59:        cmdReadFile.addActionListener(new  ActionListener() {
60:              public void actionPerformed (ActionEvent evt) {
61:                  cmdReadFileActionPerformed(evt);
62:              }
63:          });
64:          cmdCal = new  JButton("Calculate");
65:        cmdCal.setForeground(Color.blue);
66:
67:          cmdCal.addActionListener(new  ActionListener() {
68:              public void actionPerformed (ActionEvent evt) {
69:                  cmdCalActionPerformed(evt);
70:              }
71:          });
72:        cmdClear = new  JButton("Clear Data");
```

```
 73:        cmdClear.setForeground(Color.blue);
 74:
 75:          cmdClear.addActionListener(new   ActionListener() {
 76:              public void actionPerformed( ActionEvent evt) {
 77:                  cmdClearActionPerformed(evt);
 78:              }
 79:          });
 80:        cmdPrint = new  JButton("Print Result");
 81:        cmdPrint.setForeground(Color.blue);
 82:
 83:        cmdPrint.setEnabled(false);
 84:          cmdPrint.addActionListener(new   ActionListener() {
 85:              public void actionPerformed( ActionEvent evt) {
 86:                  cmdPrintActionPerformed(evt);
 87:              }
 88:          });
 89:        buttonPanel = new JPanel();
 90:        buttonPanel.add(cmdReadFile);
 91:        buttonPanel.add(cmdCal);
 92:        buttonPanel.add(cmdClear);
 93:        buttonPanel.add(cmdPrint);
 94:
 95:              /* statistical display */
 96:        meanLabel = new JLabel("                          ");
 97:        sdLabel = new JLabel("                            ");
 98:        statPanel = new JPanel();
 99:        statPanel.add(meanLabel);
100:        statPanel.add(sdLabel);
101:        statPanel.setVisible(false);
102:
103:              /* table and tablePanel */
104:  tablePanel = new JPanel(new BorderLayout());
105:  table = new JTable(new TScoreTableModel());
106:  table.setPreferredScrollableViewportSize(new
          Dimension(400,200));
107: jScrollPane2 = new JScrollPane(table);
108:  tablePanel.add(jScrollPane2);
109:  tablePanel.setVisible(false);
110:
111:        /* put them all together in leftPanel */
112:  leftPanel = new JPanel();
113:  leftPanel.setBorder(BorderFactory.createRaisedBevelBorder());
114:        leftPanel.add(inputLabel);
115:        leftPanel.add(jScrollPane1);
116:        leftPanel.add(buttonPanel);
117:        leftPanel.add(statPanel);
118:        leftPanel.add(tablePanel);
119:
120:        /* create gradeOptionPanel, helpPanel and aboutPanel
121:        * and keep them in TabbedPane
122:        */
123:        gradeOptionPanel = new GradeOptionPanel();
124:        helpPanel = new HelpPanel();
125:        aboutPanel = new AboutPanel();
126:        tabPane = new JTabbedPane();
127:        tabPane.setForeground(Color.blue);
128:        tabPane.addTab("Grade Option",gradeOptionPanel);
129:        tabPane.addTab("User Guide",helpPanel);
130:         tabPane.addTab("About", aboutPanel);
131:
```

```
132:        /* create subMainPanel which consists of leftPanel on the
left
133:        * and TabbedPane on the right side
134:        */
135:        subMainPanel = new JPanel( new GridLayout(0,2));
136:        subMainPanel.add(leftPanel);
137:        subMainPanel.add(tabPane);
138:
139:        /* Now add all panels into this TScorePanel */
140:
141:        setLayout(new BorderLayout());
142:        add(headerPanel, BorderLayout.NORTH);
143:        add(subMainPanel, BorderLayout.CENTER);
144:      }
145:
146:      private void cmdCalActionPerformed( ActionEvent evt) {
147:
148:        numberOfData= scoreText.readScore();
149:    Score = scoreText.getScore();
150:    if (numberOfData > 0) {
151:          gradeOption = gradeOptionPanel.getGradeOption();
152:          tablePanel.setVisible(true);
153:          statPanel.setVisible(true);
154:          cmdPrint.setEnabled(true);
155:          TScoreTableModel ttm = new TScoreTableModel(
                   Score, numberOfData, gradeOption);
156:          table.setModel(ttm);
157:
158:              // Table cell alignment
159:          TableColumnModel tcm = table.getColumnModel();
160:
161:          DefaultTableCellRenderer cellAlignmentCenter
                 = new DefaultTableCellRenderer();
162:  cellAlignmentCenter.setHorizontalAlignment(
                 SwingConstants.CENTER);
163:          tcm.getColumn(1).setCellRenderer(cellAlignmentCenter);
164:          tcm.getColumn(2).setCellRenderer(cellAlignmentCenter);
165:          tcm.getColumn(5).setCellRenderer(cellAlignmentCenter);
166:
167:          DefaultTableCellRenderer cellAlignmentRight = new
                 DefaultTableCellRenderer();
168:          cellAlignmentRight.setHorizontalAlignment(
                 SwingConstants.RIGHT);
169:          tcm.getColumn(3).setCellRenderer(cellAlignmentRight);
170:          tcm.getColumn(4).setCellRenderer(cellAlignmentRight);
171:
172:          avg = ttm.getMean();
173:          sd = ttm.getSD();
174:
175:        meanLabel.setText("Average raw  score  : "+
          String.format("%.2f",avg));
176:        sdLabel.setText("  Standard Deviation. :
          "+String.format("%.2f", sd));
177:    } else {
178:    if (numberOfData == 0)
179:    JOptionPane.showMessageDialog(null,"Data not found!! \n
                  Please input your scores!! ",
                 "Warning",JOptionPane.WARNING_MESSAGE);
180:      }
181:  }
182:      private void cmdClearActionPerformed( ActionEvent evt) {
```

```
183:  cmdPrint.setEnabled(false);
184:  scoreText.setText("");
185:      }
186:      private void cmdReadFileActionPerformed( ActionEvent evt) {
187:  JFileChooser chooser = new JFileChooser();
188:       chooser.setMultiSelectionEnabled(false);
189:       int chooserSelection = chooser.showOpenDialog(this);
190:       if (chooserSelection == JFileChooser.APPROVE_OPTION) {
191:        // Read in data file to ScoreTextArea
192:              try{
193:              String strLine;
194:        fileName = "";
195:        fileName = chooser.getSelectedFile().toString();
196:        FileInputStream inStream = new
           FileInputStream(chooser.getSelectedFile());
197:        BufferedReader  bufferedData = new
           BufferedReader(new InputStreamReader(inStream));
198:        while ((strLine = bufferedData.readLine()) != null) {
199:              scoreText.append(strLine + "\n");
200:        }
201:              }catch(Exception e){
202:        JOptionPane.showMessageDialog(null,"Cannot read  file  "+
           chooser.getSelectedFile(),
           "Error",JOptionPane.WARNING_MESSAGE);
203:              }
204:       }
205:       else {
206:        JOptionPane.showMessageDialog(null,"File Selection
      canceled","Information",JOptionPane.INFORMATION_MESSAGE);
207:       }
208:      }
209:      private void cmdPrintActionPerformed( ActionEvent evt) {
210:
211:  PrinterJob prntJob = PrinterJob.getPrinterJob();
212:  prntJob.setPrintable(new Printable() {
213:  int maxPage=1;
214:
215:  public int print (Graphics g, PageFormat pageFormat,
      int pageIndex) {
216:              int r,c ; // row and column in printable papaer
217:              Date today = new Date();
218:
219:   if(pageIndex >= maxPage ) {
220:        return NO_SUCH_PAGE;
221:   }    else{
222:  Graphics2D g2 = (Graphics2D) g;
223:  g2.translate(pageFormat.getImageableX(),
           pageFormat.getImageableY());
224:  pageFormat.setOrientation(pageFormat.PORTRAIT);
225:  int widthPage= (int)pageFormat.getImageableWidth();
226:  int heightPage =(int)pageFormat.getImageableHeight();
227:
228:  int y =20;
229:  if ( pageIndex > 0) g2.drawString("page : "
           +( pageIndex+1)+" / "+maxPage,300, y) ;
230:         y+=20;
231:         g2.setFont( new Font("Dialog", Font.BOLD,10));
232:         FontMetrics fontMetrics = g2.getFontMetrics();
233:         y += fontMetrics.getAscent();
234:         g2.drawString( "Normalized T Score
                   Calculation Report ", 200,y);
```

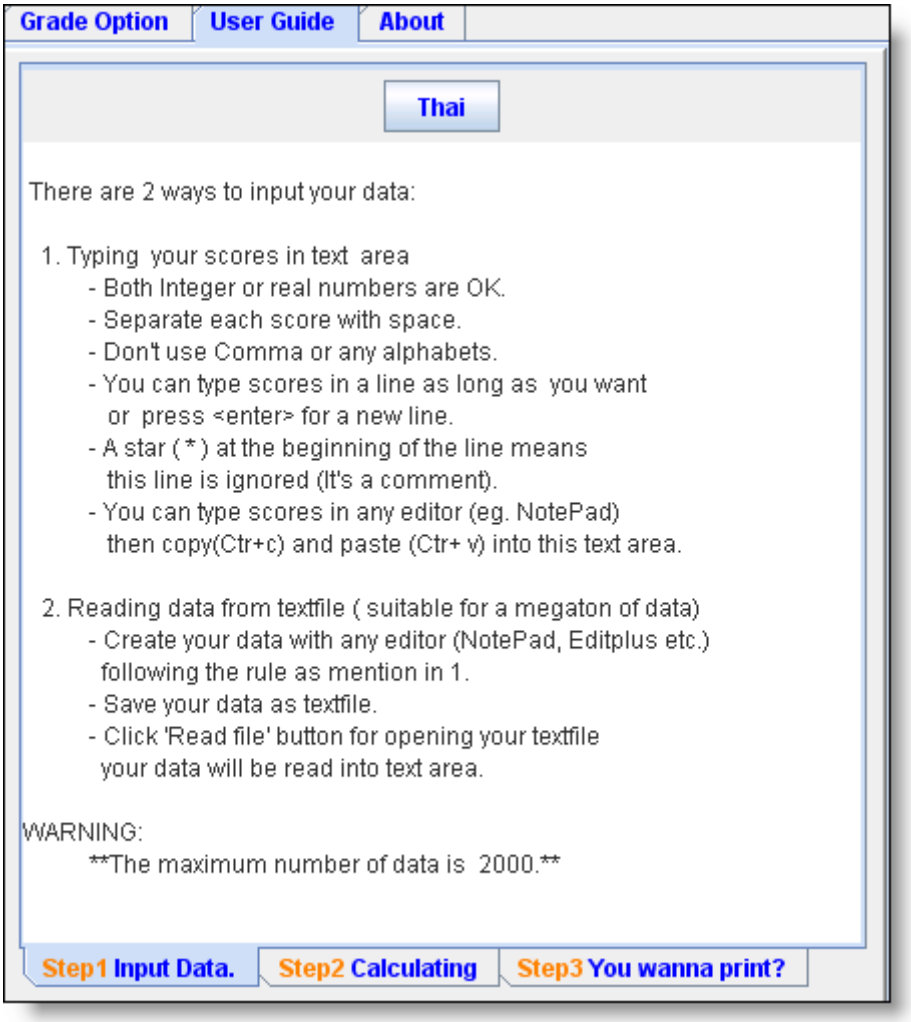
```
235:         y +=20;
236:         g2.drawString("Data from  :  " + fileName, 120,y);
237:         y+=20;
238:         g2.drawString("Printed Date  : "+ today,120,y);
239:         y+=20;
240:        g2.drawString("*** " +meanLabel.getText(), 120, y );
             // message, column, row
241:        g2.drawString(sdLabel.getText(), 270,y);
242:        y+=10;
243:        g2.drawLine(120,y, 540,y);
244:        y+=10;
245:
246:        TableColumnModel colModel = table.getColumnModel();
247:        int numCols = colModel.getColumnCount();
248:        int [ ] x = new int [ numCols];
249:        x[0] =120;
250:         y += g2.getFontMetrics().getHeight();
251:         for (c=0; c < numCols ; c ++) {
252:              TableColumn tableCol = colModel.getColumn(c);
253:              int width = tableCol.getWidth();
254:              if (c+1 < numCols) x[c+1] = x[c] + width;
255:              String headTitle =(String)tableCol.getIdentifier();
256:              g2.drawString(headTitle, x[c], y);
257:        }
258:        //g2.setFont(table.getFont());
259:        g2.setFont( new Font("Dialog", Font.PLAIN,8));
260:        int head = y;
261:        int h = g2.getFontMetrics().getHeight();
262:  //    int  tableRowHeight = max((int)(h*1.5),10);
263:        int  tableRowHeight = max((int)(h*1.1),10);
264:        int rowPerPage = (heightPage -head)/ tableRowHeight;
265:        maxPage = max((int)ceil(table.getRowCount()/
            (double)rowPerPage),1);
266:
267:        TableModel tModel = table.getModel();
268:        int startRow = pageIndex*rowPerPage;
269:        int endRow = min(table.getRowCount(),
                   startRow+rowPerPage);
270:
271:        for  (r = startRow; r < endRow; r++) {
272:              y += h;
273:              for(c=0;  c < numCols; c++) {
274:              int col = table.getColumnModel().getColumn(c).
                        getModelIndex();
275:              Object cell = tModel.getValueAt(r,col);
276:              String str = cell.toString();
277:              g2.drawString(str, x[c]+20, y);
278:              } // for c
279:              } // for r
280:              return PAGE_EXISTS;
281:        }
282:        } // method print
283:  } // new printable
284:        );
285:  if (prntJob.printDialog())    {
286:                    try{
287:                          prntJob.print();
288:                          } catch(Exception e) {
289:                    JOptionPane.showMessageDialog(null,
      "Printer Error","Error",JOptionPane.WARNING_MESSAGE);
290:              }
```

```
291:  }
292:
293:      }
294:
295:  private  JLabel headerLabel;
296:  private JLabel headerLabel2;
297:  private  JPanel headerPanel;
298:
299:  private  ScoreTextArea scoreText;
300:  private  JScrollPane jScrollPane1;
301:
302:  private  JButton cmdReadFile;
303:  private  JButton cmdCal;
304:  private  JButton cmdClear;
305:  private JButton cmdPrint;
306:  private JPanel buttonPanel;
307:
308:  private JLabel meanLabel ;
309:  private     JLabel sdLabel;
310:  private JPanel statPanel;
311:
312:  private  JTable table;
313:  private JPanel  tablePanel;
314:  private JScrollPane jScrollPane2;
315:
316:  private  JPanel leftPanel;
317:
318:  private HelpPanel helpPanel;
319:  private GradeOptionPanel gradeOptionPanel;
320:  private AboutPanel aboutPanel;
321:  private JTabbedPane tabPane;
322:
323:  private JPanel subMainPanel;
324:
325:  private int numberOfData;
326:  private double [ ] Score = new double[MAXIMUM_NUMBER_OF_DATA];
327:  private String fileName=" text area by user";
328:  private int gradeOption=0;
329:  //  statistical value for printing
330:  private double avg, sd;  // mean and standard deviation
331:
332:  }
333:
```

เมื่อผู้ใช้คลิกปุ่ม Calculate ผลการคำนวณ T ปกติ พร้อมเกรด จะแสดงให้เห็นในตาราง ในขณะที่ ปุ่ม Print result จะ enabled สามารถคลิกเพื่อส่งผลลัพธ์พิมพ์ออกทางเครื่องพิมพ์ได้ เมธอดที่ทำหน้าที่พิมพ์ อยู่ที่บรรทัด 209 ถึง 284 โปรแกรมจะสร้างวัตถุที่ชื่อว่า prntJob จากคลาส PrinterJob และเรียกใช้งานคลาส Printable ซึ่งมีเมธอด print ( ) ซึ่งทำหน้าที่สร้างเอกสารแต่ละหน้าสำหรับพิมพ์ ในการพัฒนาโปรแกรมนี้ ได้สร้างคลาส Printable ซ้อนเป็นคลาสย่อยในคลาส TScorePanel ซึ่งจะสะดวกในการเรียกใช้ตัวแปรและแก้ไขโปรแกรม

4.5 ส่วนที่แสดงข้อความแนะนำการใช้งานโปรแกรม

คลาสที่ทำหน้าที่แสดงข้อความแนะนำการใช้งานโปรแกรมคือ คลาส HelpPanel จะถูกวางซ้อนเป็นแผ่นที่ 2 ชื่อ User Guide ในแท็บที่อยู่ด้านขวามือ คำแนะนำการใช้งานมีทั้งที่เป็นภาษาไทยและภาษาอังกฤษ สามารถสลับภาษาได้โดยการใช้เมาส์คลิกปุ่มเลือกภาษาที่อยู่ด้านบน



รูปที่ 4.5 แสดง Panel ที่แนะนำวิธีการใช้โปรแกรม

คำแนะนำวิธีใช้งานจะแบ่งออกเป็น 3 ขั้นตอน คือ ขั้นตอนการป้อนข้อมูล ขั้นตอนการประมวลผล และ ขั้นตอนการส่งผลลัพธ์ออกไปสู่เครื่องพิมพ์

Listing 4.9 แสดงรายละเอียดของคลาส HelpPanel

```
1:  /* File :HelpPanel.java
2:  * Project: Math Tools in Java
3:  * Purpose: Show how to use Normalized T score calculation
4:  * Author : Wachara R.
```

```
 5:  * First Released: Fri 23 May 2008
 6:  * Last Updated :  Tu 17 June 2008
 7:  */
 8:  package JApp.TS;
 9:  import javax.swing.*;
10:  import java.awt.*;
11:  import java.awt.event.*;
12:
13:  public class HelpPanel extends JPanel {
14:  JTabbedPane userGuidePane = new TabbedPane(JTabbedPane.BOTTOM);
15:  UserGuidePanel  tab1;
16:  UserGuidePanel  tab2;
17:  UserGuidePanel  tab3;
18:
19:  String str1 =
20:        "\n There are 2 ways to input your data: \n\n
      1. Typing  your scores in text  area \n\t"+
21:        "- Both Integer or real numbers are OK.\n\t" +
22:        "- Separate each score with space.\n\t- Don't
           use Comma or any alphabets.\n\t"+
23:        "- You can type scores in a line as long as
           you want\n\t"+
24:        "   or  press <enter> for a new line.
     \n\t- A star ( * ) at the beginning of the line means\n\t "+
25:        "  this line is ignored (It's a comment). \n\t"+
26:        "- You can type scores in any editor (eg. NotePad)\n\t" +
27:        "   then copy(Ctr+c) and paste (Ctr+ v) into
            this text area.\n\n" +
28:        "   2. Reading data from textfile ( suitable for
            a megaton of data)\n\t"+
29:        "- Create your data with any editor (NotePad,
           Editplus etc.)\n\t"+
30:        "  following the rule as mention in 1.\n\t"+
31:        "- Save your data as textfile. \n\t"+
32:        "- Click 'Read file' button for opening
           your textfile\n\t"+
33:        "  your data will be read into text area.\n\n"+
34:        "WARNING:\n"+
35:        "\t**The maximum number of data is  2000.** ";
36:  String str2 =
37:        "\n\t- When data appear in text area,
           clicking on 'Calculate' button\n"+
38:        "  will launch the program crunching the numbers.
            Sorted scores, frequency\n"+
39:        "  T score and grade  being displayed in the
            table in no time\n\n"+
40:        "\t- Selecting data in table (by draging
           mouse in table area), \n"+
41:        "  You can copy (Ctr+c) and paste (Ctr+v)
           them into word  processing \n"+
42:        "   or spread sheet application.\n\n"+
43:        "\t- To change the range of grade assesment,
            selecting any\n "+
44:        "specific range as shown in the grade option
           panel and click on \n"+
45:        "Calculate'  button  for recalculation.";
46:  String str3 =
47:        "\n\n\t- Just clicking on 'Print result' button,
            the output result will be\n"+
48:        "sent to the printer\n\n"+
49:        " WARNING ..\n\n"+
```

```
50:         "\tThe paper size setting is limited to A4
            ( 11\" x 8.5\" ) \n";
51:
52: String Tstr1 = "\n ผู้ใช้สามารถป้อนข้อมูลได้ 2 วิธี: \n\n
    1. พิมพ์คะแนนลงไปในช่อง text area \n\t"+
53:         "- คะแนนจะเป็นจำนวนเต็มหรือจำนวนที่มีทศนิยมก็ได้.\n\t" +
54:         "- แยกคะแนนแต่ละคะแนนด้วย 'ช่องว่าง'.\n\t-
    ห้ามใช้เครื่องหมายจุลภาค หรือ ตัวอักขระใด ๆ.\n\t"+
55:         "- บรรทัดหนึ่ง ๆ จะพิมพ์คะแนนกี่ค่าก็ได้ตามแต่จะต้องการ \n\t"+
56:         "  ต้องการขึ้นบรรทัดใหม่ให้กดปุ่ม 'enter'. \n\t-
    ต้องการเขียน comment ให้ใส่ * ที่หน้าบรรทัดนั้น\n\t "+
57:         " บรรทัดนั้นจะไม่ถูกนำไปประมวลผล. \n\t"+
58:         "- เราอาจใช้โปรแกรม editor เช่น NotePad พิมพ์คะแนน
    เตรียมไว้ก่อน\n\t" +
59: "  จากนั้น copy(Ctr+c) แล้ว paste (Ctr+ v)มาใส่ใน text area.\n\n" +
60:         "    2. อ่านค่าคะแนนจาก text file (เหมาะสำหรับกรณีที่มี
    ข้อมูลเป็นจำนวนมาก)\n\t"+
61: "- พิมพ์คะแนนลงในโปรแกรม editor ใด ๆ (NotePad, Editplus ฯลฯ) \n\t"+
62: "  กฎเกณฑ์การพิมพ์คะแนนเหมือนกับที่กล่าวไว้ในข้อ 1.\n\t"+
63:         "- บันทึกข้อมูล เก็บไว้เป็น text file. \n\t"+
64:         "- คลิกปุ่ม 'Read file' เพื่ออ่านคะแนนจากแฟ้มข้อมูลนั้น\n\t"+
65: "  คะแนนที่ถูกเก็บไว้ในแฟ้มข้อมูลจะถูกนำมาแสดงใน text area.\n\n"+
66:         "คำเตือน:\n"+
67:         "\t**จำนวนข้อมูลหรือคะแนนต้องไม่เกิน 2000 ค่า** ";
68: String Tstr2 = "\n\t- เมื่อข้อมูลถูกแสดงใน text area ,
            ให้คลิกที่ปุ่ม'Calculate'\n"+
69:         "  โปรแกรมจะทำการประมวลผล โดยจะเรียงลำดับข้อมูล หาความถี่\n"+
70:         "  คำนวณ T Score และกำหนดระดับคะแนน แสดงให้เห็นในตาราง
    ชั่วอึดใจ\n\n"+
71: "\t- สามารถเลือกข้อมูลในตาราง (โดยลากเมาส์ระบายในบริเวณตาราง), \n"+
72:         " จากนั้น copy (Ctr+c) และ paste (Ctr+v) ข้อมูลไปใช้ใน
73: โปรแกรม word processing \n"+
```

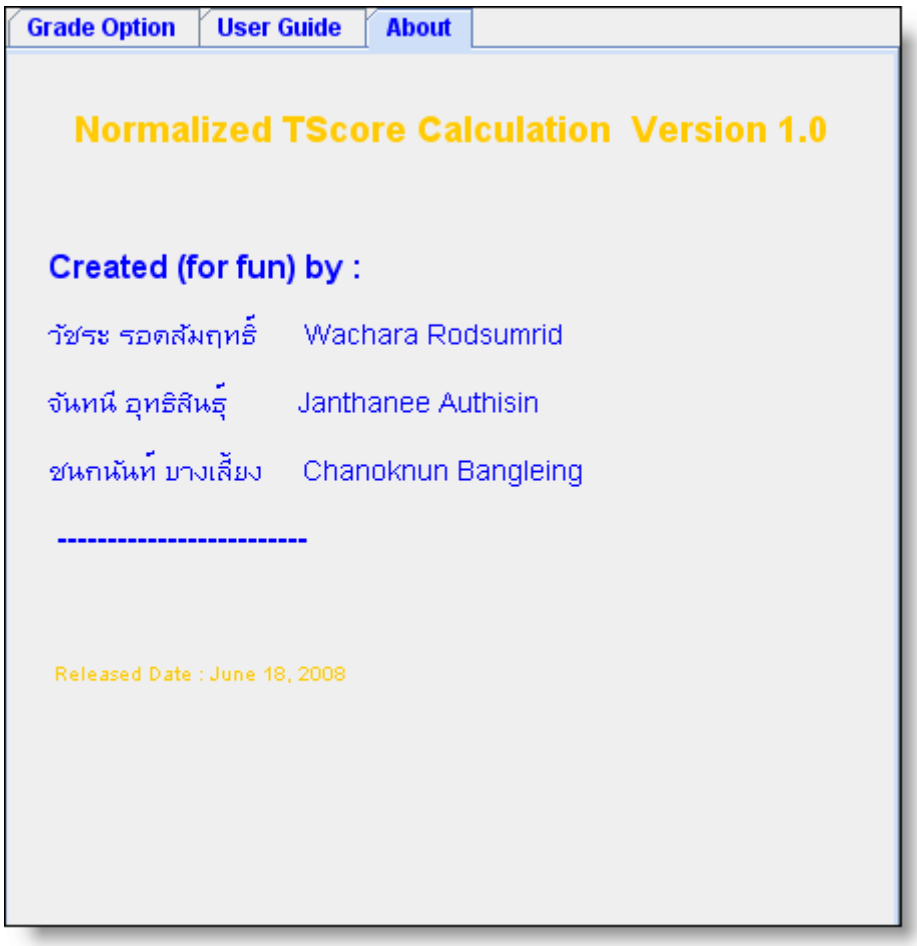
```
74:         "    หรือโปรแกรมตารางคำนวณ.\n\n"+
75:  "\t- ต้องการเปลี่ยนช่วงของระดับคะแนน, ให้คลิกเลือกช่วงระดับคะแนน\n "+
76:  "ที่ต้องการดังปรากฏอยู่ที่ช่องด้านขวามือ จากนั้นคลิกที่ปุ่ม 'Calculate'  \n"+
77:         "เพื่อให้โปรแกรมทำการคำนวณใหม่อีกครั้ง.";
78:  String Tstr3 = "\n\n\t- ต้องการพิมพ์ผลลัพธ์ให้คลิกปุ่ม 'Print result'
79: ข้อมูลในตาราง\n"+
80:         "จะถูกส่งไปพิมพ์ที่เครื่องพิมพ์\n\n"+
81:         " คำเตือน ..\n\n"+
82:"\t**ได้กำหนดกระดาษเป็นค่ามาตรฐานไว้ที่ขนาด A4 ( 11\" x 8.5\" )**  \n";
83:  public HelpPanel()
84:  {
85:   tab1=new UserGuidePanel(userGuidePane,str1, Tstr1);
86:  tab2=new UserGuidePanel(userGuidePane,str2,Tstr2);
87:  tab3=new UserGuidePanel(userGuidePane,str3,Tstr3);
88:  setBorder(BorderFactory.createRaisedBevelBorder());
89:  userGuidePane.setForeground(Color.blue);
90:  userGuidePane.addTab("<html><font color=#FF8000>Step1
     </font>Input Data.</html> ", tab1);
91:  userGuidePane.addTab("<html><font color=#FF8000>Step2
     </font> Calculating</html> ", tab2);
92:  userGuidePane.addTab("<html><font color=#FF8000>Step3
     </font>You wanna print?</html> ", tab3);
93:  this.add(userGuidePane);
94:
95:  }
96:
97:  }
98:
```

aboutPanel เป็น คลาสที่ใช้แสดงรุ่นของโปรแกรม ชื่อผู้พัฒนาโปรแกรม และวันที่เริ่มใช้โปรแกรมนี้ วางซ้อนเป็นแผ่นที่ 3 ชื่อ About ในแท็บ จากรูปจะเห็นว่า ผู้พัฒนาได้กำหนดให้โปรแกรมที่กำลังใช้งานนี้เป็นเวอร์ชัน 1.0 เริ่มเผยแพร่และใช้งานเมื่อวันที่ 18 มิถุนายน 2551

Listing 4.10 แสดงรายละเอียดของคลาส AboutPanel

```
1:    /* File :AboutPanel.java
  2:  * Project: Math Tools in Java
  3:  * Purpose: Display History and staff
  4:  * Author : Wachara R.
  5:  * First Released: Tu 3 June 2008
  6:  * Last Updated :  Wed 18 June  2008
  7:  */
  8:  package JApp.TS;
  9:  import javax.swing.*;
 10:  import java.awt.*;
```

```
11:  import java.awt.event.*;
12:
13:  public class AboutPanel extends JPanel {
14:  JLabel jLabel1 = new JLabel("Normalized TScore Calculation
           Version 1.0", JLabel.CENTER);
15:  JLabel jLabel2 = new JLabel("    ");
16:  JLabel jLabel3 = new JLabel("Created (for fun) by :");
17:  JLabel jLabel4 = new JLabel("วัชระ รอดสัมฤทธิ์   Wachara Rodsumrid");
18:  JLabel jLabel5 = new JLabel("จันทนี อุทธิสินธุ์ Janthanee Authisin");
19:  JLabel jLabel6 = new JLabel("ชนกนันท์ บางเลี้ยง Chanoknun
                      Bangleing");
20:  JLabel jLabel7 = new JLabel(" ------------------------");
21:  JLabel jLabel8 = new JLabel ( "  ");
22:  JLabel jLabel9 = new JLabel(" Released Date : June 18, 2008");
23:
24:
25:  public AboutPanel()
26:  {
27:  setLayout( new GridLayout(15,0));
28:  setBorder(BorderFactory.createEmptyBorder(20,20,10,20));
                        // top, left , bottom, right
29:  jLabel1.setForeground(Color.orange);
30:  jLabel1.setFont(new Font("Dialog", Font.BOLD, 18));
31:  jLabel3.setForeground(Color.blue);
32:  jLabel3.setFont(new Font("Dialog", Font.BOLD, 16));
33:  jLabel4.setForeground(Color.blue);
34:  jLabel4.setFont(new Font("Dialog", Font.PLAIN, 14));
35:  jLabel5.setForeground(Color.blue);
36:  jLabel5.setFont(new Font("Dialog", Font.PLAIN, 14));
37:  jLabel6.setForeground(Color.blue);
38:  jLabel6.setFont(new Font("Dialog", Font.PLAIN, 14));
39:  jLabel7.setForeground(Color.blue);
40:  jLabel7.setFont(new Font("Dialog", Font.BOLD, 14));
41:  jLabel9.setForeground(Color.orange);
42:  jLabel9.setFont(new Font("Dialog", Font.PLAIN, 10));
43:
44:  add(jLabel1); add(jLabel2); add(jLabel3);
45:  add(jLabel4); add (jLabel5); add(jLabel6);
                 add(jLabel7);add(jLabel8);add(jLabel9);
46:
47:  }
48:  }
```



รูปที่ 4.6 แสดงภาพ Panel ที่ใช้บอก version ของโปรแกรม

4.6 การนำโปรแกรมไปติดตั้งที่แม่ข่ายอินเตอร์เน็ต

เมื่อพัฒนาและทดสอบโปรแกรมว่าได้ทำงานบรรลุตามวัตถุประสงค์แล้ว ต่อไปเป็นขั้นตอนการนำโปรแกรมไปติดตั้งที่แม่ข่ายอินเตอร์เน็ต แม่ข่ายอินเตอร์เน็ตหรือ Server ที่จะนำไปติดตั้งนี้เป็นของสาขาวิชาฟิสิกส์ มีหมายเลข IP คือ 203.158.100.140

การติดตั้งโปรแกรมจะแบ่งเป็น 2 แบบ คือ แบบที่ให้โปรแกรมทำงานในลักษณะแอพเพล็ตและแบบที่ให้โปรแกรมทำงานเป็น desktop Application ด้วย Java Web Start

4.6.1 ติดตั้งให้โปรแกรมทำงาน แบบ Applet ดำเนินการตามลำดับขั้นดังนี้

1. สร้างคลาสที่สืบทอดมาจาก JApplet เพื่อเป็นทางเข้าหรือจุดเริ่มต้นสำหรับการทำงานแบบแอพเพล็ตคลาสนี้มีความยาวไม่มากนัก ดังรายละเอียดต่อไปนี้

Listing 4.11 แสดงรายละเอียดคลาส TScoreApplet

```
1: import javax.swing.JApplet;
2: public class TScoreApplet extends JApplet {
```

```
3:      public void init() {
4:    add(new TScorePanel());
5:    }
6:  }
```

2. รวม class file ทั้งหลายเข้าด้วยกัน แล้วบีบอัดให้มีขนาดเล็กลงเป็น Jar file โดยใช้คำสั่งดังนี้

```
jar –cvf TScoreApplet.jar myManifest.txt AppletClass/TScore2
MathTools/Common MathTools/Integration MatTools/SimpleStat
```

Option ที่ตามหลังคำสั่ง jar นั้น มีความหมายดังนี้

c หมายถึง ทำให้ไฟล์ถูกบีบอัด หรือ compact

v หมายถึงให้แสดงผลการทำงานของ jar อย่างละเอียด

f หมายถึงให้นำคลาสไฟล์ที่บีบอัดแล้วนั้น รวมเป็น jar file ที่มีชื่อว่า TscoreApplet.jar จากนั้นตามด้วยชื่อโฟลเดอร์ ที่เราเก็บคลาสไฟล์ ต่าง ๆ ไว้ ตามหมวดหมู่ ในที่นี้ได้แก่ AppletClass/TScore2 MathTools/Common MathTools/Integration MathTools/SimpleStat

3. สร้างเอกสาร html ในที่นี้ชื่อ TScore2.html ในเอกสารนี้จะมีการเรียก applet ชื่อ TScoreApplet.jar ขึ้นมาทำงาน เนื้อหาในเอกสารมีดังนี้

รายละเอียดในเอกสาร ชื่อ TScore2.html

```
<HTML>
<HEAD>
<TITLE> TScore Test, an improved one</TITLE>
</HEAD>
<BODY>
<center>
<Applet  codebase="./ " archive="./TScoreApplet.jar"
code="AppletClass.TScore2.TScoreApplet.class" width=900
height=650></Applet>
</center>
</BODY>
</HTML>
```

บรรทัดที่พิมพ์ด้วยอักษรตัวหนา คือหัวใจสำคัญของไฟล์นี้ tag <Applet > </Applet> เป็นการเรียกใช้งานแอพเพล็ต ระหว่าง tag นี้มี ตัวเลือก หรือ option ต่าง ๆ ซึ่งมีความหมายดังนี้

codebase เป็นการระบุ base URL ของ applet class ที่จะถูกเรียกมาทำงาน หากไม่ระบุ option นี้ URL ของไฟล์ HTMLที่เรียกใช้แอพเพล็ตนี้ จะถูกใช้เป็น base URL

archives เป็นการระบุชื่อของ jar file ที่มีแอพเพล็ตที่จะถูกเรียกทำงาน ใช้ในกรณีที่มีการบีบอัดแอพเพล็ต มาในรูป jar file

code เป็นการระบุชื่อคลาสที่จะใช้เป็นคลาสเริ่มต้นในการทำงาน

width และ height ระบุความกว้างและความสูงของแอพเพล็ต ที่จะแสดงผลในบราวเซอร์ มีหน่วยเป็น pixel

4.6.2 แบบที่ให้โปรแกรมทำงานเป็น desktop Application ด้วย Java Web Start

1. สร้างคลาส TScoreApp สืบทอดมาจาก JFrame ให้คลาสนี้มีเมธอด main ( ) เพื่อเป็นทางเข้าหรือจุดเริ่มต้นสำหรับการทำงาน

Listin 4.12 แสดงรายละเอียดของคลาส TScoreApp

```
 1: package JApp.TS;
 2: import javax.swing.*;
 3: import java.awt.*;
 4: import java.awt.event.*;
 5:
 6: public class TScoreApp extends JFrame {
 7:
 8: public TScoreApp() {
 9:   super("Normalized TScore Calculation ");
10:   setSize(900, 650);
11:   setDefaultCloseOperation(EXIT_ON_CLOSE);
12:    setResizable(false);
13:   Container content = getContentPane();
14:   //content.setLayout(new FlowLayout());
15:   content.add(new TScorePanel());
16: }
17:    public static void main(String args[]) {
18:      TScoreApp ts = new TScoreApp();
19:       ts.setVisible(true);
20:    }
21:  }
```

2. รวม class file ทั้งหลายเข้าด้วยกัน แล้วบีบอัดให้มีขนาดเล็กลงเป็น jar file ก่อนที่จะรวมต้องสร้าง Manifest file ซึ่งเป็นไฟล์ที่จะบอกให้ JRE รู้ว่าจะเริ่มทำงานโดยเรียกใช้คลาสใดก่อน

ให้ชื่อ Manifest file นี้ว่า myManifest.txt มีข้อความอยู่บรรทัดเดียวดังนี้

```
Main-Class: JApp.TS.TScoreApp
```

ต่อไปใช้คำสั่ง jar เพื่อบีบอัดและรวมไฟล์ทั้งหลายเข้าด้วยกัน รวมทั้ง myManifest.txt ด้วย

```
jar –cvfm TScoreCal.jar myManifest.txt JApp MathTools/Common
MathTools/Integration MatTools/SimpleStat
```

Option m ที่เพิ่มขึ้นมาในครั้งนี้คือ ให้ jar สร้าง Manifest file โดยใช้ไฟล์ชื่อ myManifest.txt

3. สร้าง jnlp file เป็นไฟล์สำหรับบอกให้ Java Web Start รู้ว่าจะรันโปรแกรมนี้อย่างไร ไฟล์นี้เขียนด้วยภาษา XML ให้ชื่อไฟล์นี้เป็น TScoreCal.jnlp รายละเอียดของไฟล์มีดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<jnlp
  spec="1.0+"
  codebase="http://203.158.100.140/TScore"
  href="TScoreCal.jnlp">

  <information>
    <title>Normalized T Score Calculation.</title>
    <vendor>RMUTT Physics</vendor>

    <homepage href="http://203.158.100.140/TScore/index.html"/>

    <description>The utility for calculating T Score</description>
    <description kind="short">Normalized T Score Calculation via the
capabilities of JAVA WebStart.</description>
     <offline-allowed/>
  </information>
  <security>
     <all-permissions/>
  </security>
  <resources>
    <j2se version="1.6+"/>
    <jar href="TScoreCal.jar"/>
  </resources>
  <application-desc main-class="JApp.TS.TScoreApp"/>
  </jnlp>
```

4. สร้างลายเซ็น (Sign) ให้แก่ TScoreCal.jar โดยใช้เครื่องมือที่มาพร้อมกับ JDk 1.6 ได้แก่ keytool และ jarsigner keytool เป็นโปรแกรมที่จะสร้าง keystore ซึ่ง jarsigner จะรวบรวมไปเก็บไว้ใน TScoreCal.jar เมื่อ Web Start เรียก TScoreCal.jar ขึ้นมาทำงาน จะตรวจสอบข้อมูลใน jar file นี้ ว่าตรงกันกับข้อมูลที่ที่ sign ไว้หรือไม่ ทั้งนี้เพื่อความปลอดภัยในการใช้งาน ป้องกันมิให้ผู้อื่นสร้างโปรแกรมหรือแฝงคำสั่งที่ประสงค์ร้ายต่อเครื่องคอมพิวเตอร์ผนวกเข้าในโปรแกรมที่ผู้ใช้เรียกใช้ (รายละเอียดของการสร้างลายเซ็นด้วยการใช้ keytool และ jarsigner ดูได้จากภาคผนวก C)

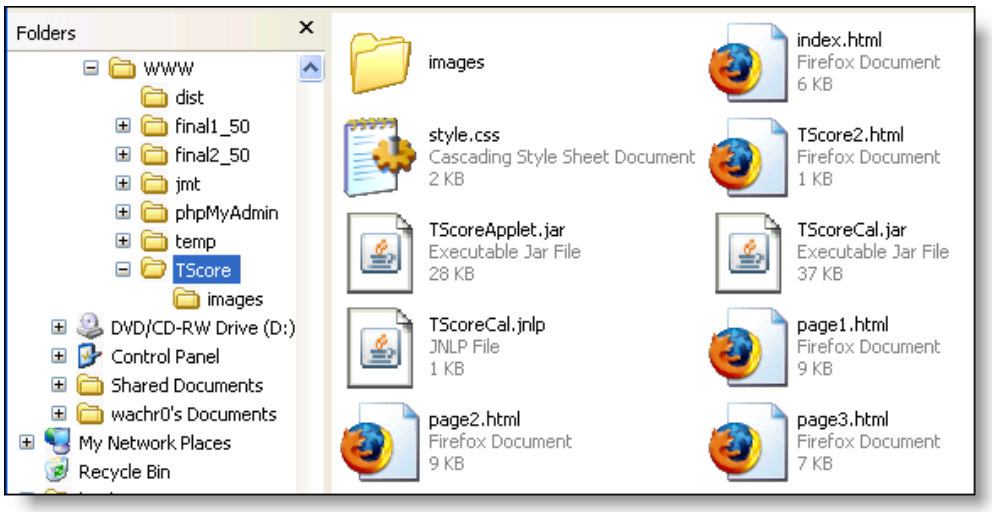
5. สร้างไฟล์ HTML ที่มีลิงค์ เรียกใช้ TScoreCal.jnlp นี้ ส่วนที่ให้ผู้ใช้งานเรียกใช้ไฟล์นี้จะมีข้อความดังนี้

.....

 **Run โปรแกรม Normalized T Score คลิกที่นี่ **
span class="style3">โปรแกรมจะถูก download แล้วนำไปติดตั้งในเครื่องคอมพิวเตอร์ของคุณ สามารถอ่านไฟล์จากฮาร์ดดิสก์ พิมพ์ผลลัพธ์ผ่านเครื่องพิมพ์ และ save โปรแกรมเก็บไว้ใช้งานภายหลังเมื่อเลิกเชื่อมต่ออินเตอร์เน็ตแล้วได้

.....

นำไฟล์ทั้งหมดนี้ไปเก็บไว้ที่ Server ที่ต้องการให้ผู้ใช้เรียกใช้งาน ในที่นี้ คือ Server ที่มีเลขไอพี 203.158.100.140 ซึ่งเป็น Server ของฟิสิกส์ราชมงคล ของสาขาวิชาฟิสิกส์ มหาวิทยาลัยเทคโนโลยีราชมงคล ธัญบุรี รวมไฟล์ทุกไฟล์เก็บไว้ภายใต้โฟลเดอร์ที่ชื่อว่า TScore



รูป 4.7 แสดงโฟลเดอร์ TScore เมื่อนำไปเก็บไว้ใน Internet Server

เมื่อผู้ใช้งานเชื่อมต่ออินเตอร์เน็ต เข้าถึงโดย http://203.158.100.140/TScore จะปรากฏหน้าแรกของเว็บไซต์ดังรูปที่ 4.8

รูปที่ 4.8 หน้าแรกของเว็บเพจที่เรียกใช้โปรแกรม

4.7 ผลการทำงานของโปรแกรมผ่านเครือข่ายอินเตอร์เน็ต

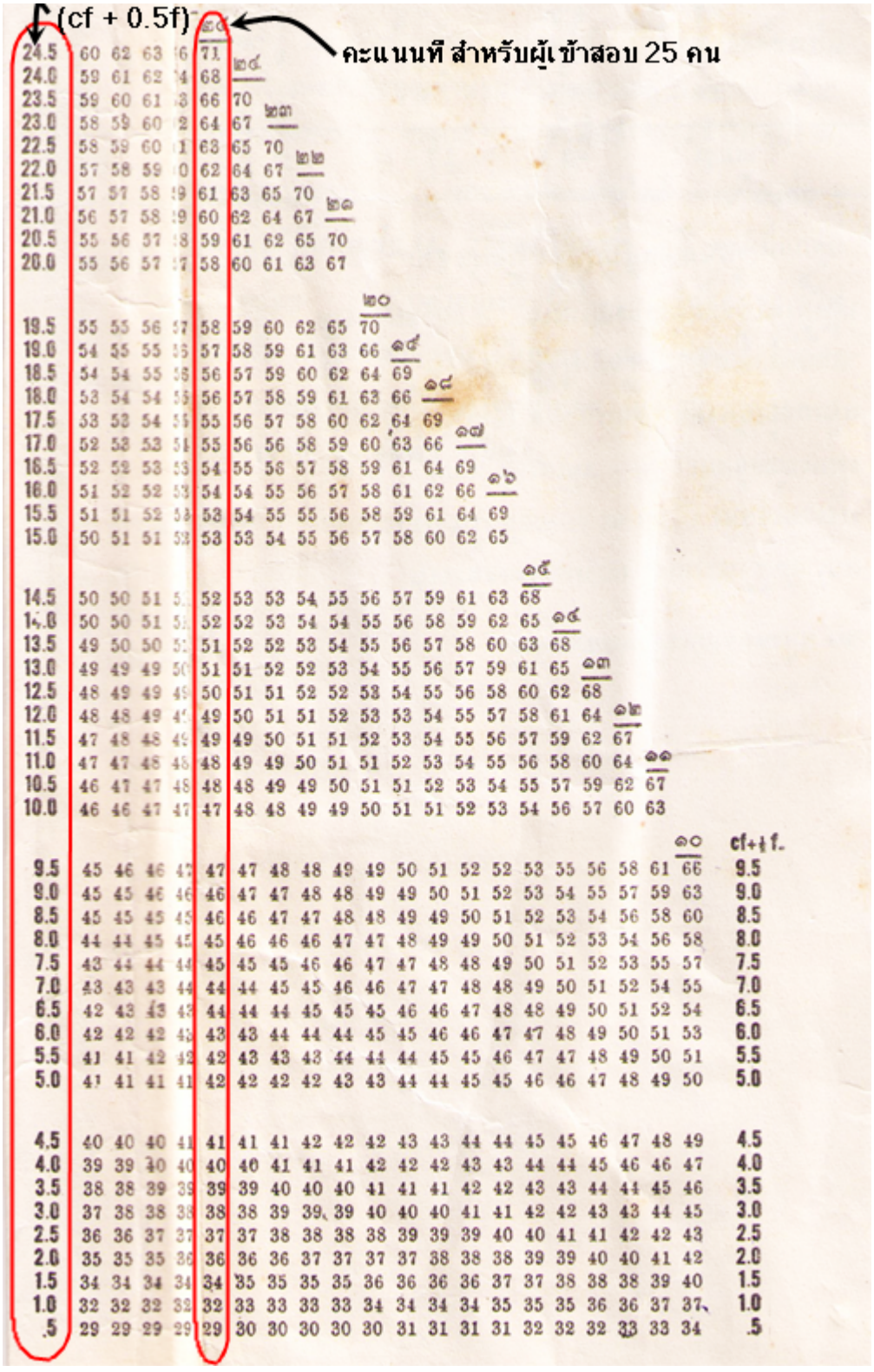
เมื่อโปรแกรมถูกเรียกใช้ผ่านเครือข่ายอินเตอร์เน็ต โปรแกรมจะถูกดาวน์โหลด มาที่เครื่องของผู้ใช้ ได้ทดสอบการทำงานของโปรแกรมเพื่อดูความถูกต้อง ดังนี้

1. ทดลองคำนวณคะแนนทีปกติโดยการเปิดตารางแล้วเปรียบเทียบค่าที่ได้กับผลลัพธ์ของโปรแกรม

ในการสอบวิชาหนึ่ง ผลการสอบของนักศึกษา จำนวน 25 คน ได้คะแนนดิบดังนี้

23 17 22 21 21 20 19 20 20 19 19 17 19 18 18 17 17 16 16 14 13 13 12 22 14

นำคะแนนดิบไปเรียงลำดับจากมากไปน้อย หาความถี่ ความถี่สะสม คำนวณหา (cf + 0.5f) แล้วนำค่านี้ไปเปิดตารางหาค่าคะแนนทีปกติ ตารางที่ใช้เปิดหาคะแนนทีปกติ แสดงค่าดังรูปที่ 4.9



รูป 4.9 ตารางคะแนนทีปกติ ในรูปนี้ใช้ได้กับผู้เข้าสอบตั้งแต่ 10 ถึง 25 คน

เมื่อคำนวณค่า (cf + 0.5f) ได้แล้วให้นำค่าที่ได้นี้ไปเทียบกับตารางตรงคอลัมน์ซ้ายมือสุด แล้วไล่ตรงมาทางขวามือ จนถึงคอลัมน์ที่มีผู้เข้าสอบ 25 คน ตัวเลขที่ได้ตรงช่องนี้คือคะแนนทีปกติที่ต้องการ

เช่น บรรทัดแรก (cf + 0.5f) เท่ากับ 24.5 เทียบแล้วตรงกับคะแนนทีปกติที่ 71

บรรทัดที่ 2 (cf + 0.5f) เท่ากับ 23 เทียบแล้วตรงกับคะแนนทีปกติที่ 64

บรรทัดที่ 2 (cf + 0.5f) เท่ากับ 21 เทียบแล้วตรงกับคะแนนทีปกติที่ 60

ทำเช่นนี้เรื่อยไป จนครบทุกค่า

นำคะแนนทีที่ได้มาจำแนกเกรด แบ่งเป็น 5 ระดับ คือ A, B, C, D และ F

พิสัยของคะแนนทีปกติคือ 71 -29 = 42 จำนวนคะแนนทีปกติในแต่ละเกรด = $\frac{42}{5}$ = 8.4 จำแนกเป็นเกรดแล้วจะได้ดังนี้

เกรด A คือคะแนนทีปกติ ตั้งแต่ 63 ขึ้นไป มีจำนวน 3 คน

เกรด B คือคะแนนทีปกติ ตั้งแต่ 55-62 มีจำนวน 5 คน

เกรด C คือคะแนนทีปกติ ตั้งแต่ 46-54 มีจำนวน 10 คน

เกรด D คือคะแนนทีปกติ ตั้งแต่ 38-45 มีจำนวน 4 คน

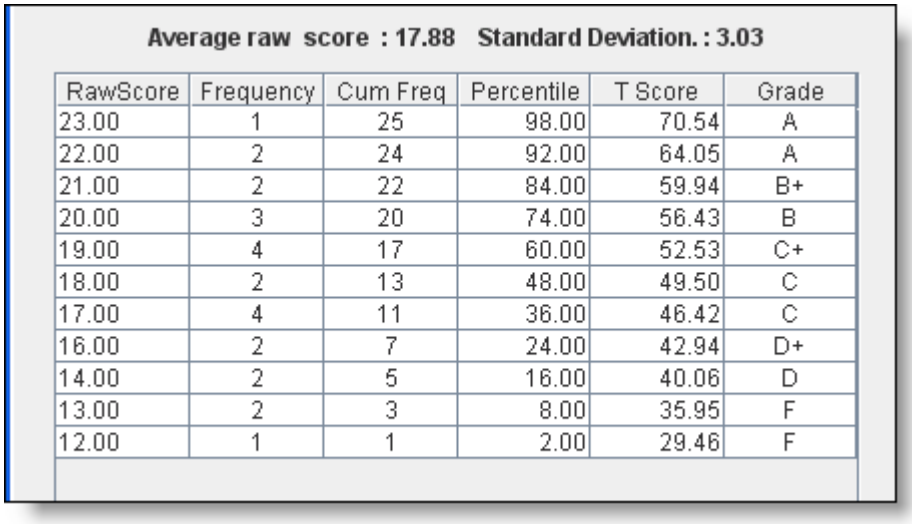
เกรด F คือคะแนนทีปกติ ตั้งแต่ 37 ลงมา มีจำนวน 3 คน

ผลลัพธ์สุดท้ายจะได้ดังรูปที่ 4.10

คะแนนดิบ	Tally	ความถี่	ความถี่สะสม	(cf+0.5f)	Percentile	T	Grade
23	/	1	25	24.5	98	71	A
22	//	2	24	23	92	64	A
21	//	2	22	21	84	60	B
20	///	3	20	18.5	74	56	B
19	////	4	17	15	60	53	C
18	//	2	13	12	48	49	C
17	////	4	11	9	36	46	C
16	//	2	7	6	24	43	D
14	//	2	5	4	16	40	D
13	//	2	3	2	8	36	F
12	/	1	1	0.5	2	29	F

รูปที่ 4.10 ผลลัพธ์ที่ได้จากการคำนวณคะแนนทีปกติโดยการเปิดตาราง

ผลลัพธ์ที่ได้จากการให้โปรแกรมประมวลผลข้อมูลดังกล่าว จะได้ดังนี้



Average raw score : 17.88 Standard Deviation. : 3.03

RawScore	Frequency	Cum Freq	Percentile	T Score	Grade
23.00	1	25	98.00	70.54	A
22.00	2	24	92.00	64.05	A
21.00	2	22	84.00	59.94	B+
20.00	3	20	74.00	56.43	B
19.00	4	17	60.00	52.53	C+
18.00	2	13	48.00	49.50	C
17.00	4	11	36.00	46.42	C
16.00	2	7	24.00	42.94	D+
14.00	2	5	16.00	40.06	D
13.00	2	3	8.00	35.95	F
12.00	1	1	2.00	29.46	F

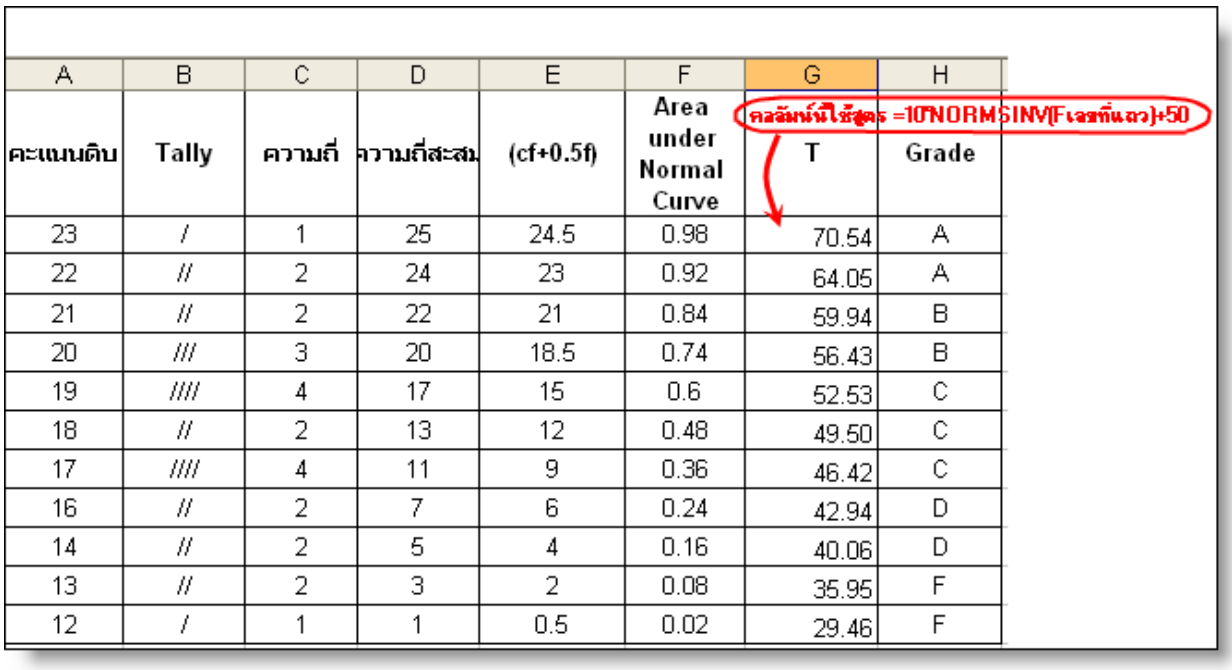
รูปที่ 4.11 แสดงผลลัพธ์ที่ได้จากโปรแกรม

ทุกคอลัมน์จะได้ค่าตรงกัน ยกเว้นตรงคอลัมน์ T Score จะต่างกันเนื่องจาก การคำนวณของโปรแกรมใช้เลขทศนิยม 2 ตำแหน่ง แต่คะแนนทีที่ได้จากการเปิดตารางเป็นเลขจำนวนเต็ม

2. ทดลองคำนวณคะแนนทีปกติโดยใช้โปรแกรม Microsoft Excel แล้วเปรียบเทียบค่าที่ได้กับผลลัพธ์ของโปรแกรม

ก. ใช้ข้อมูลชุดเดิมเหมือนกับการทดสอบในข้อที่ 1 ตรงคอลัมน์ G คำนวณหาค่า T Score โดยใช้ฟังก์ชัน NormSInv คำนวณ โดยแต่ละช่องใช้สูตร

= 10 * NormSInv (cell #) + 50



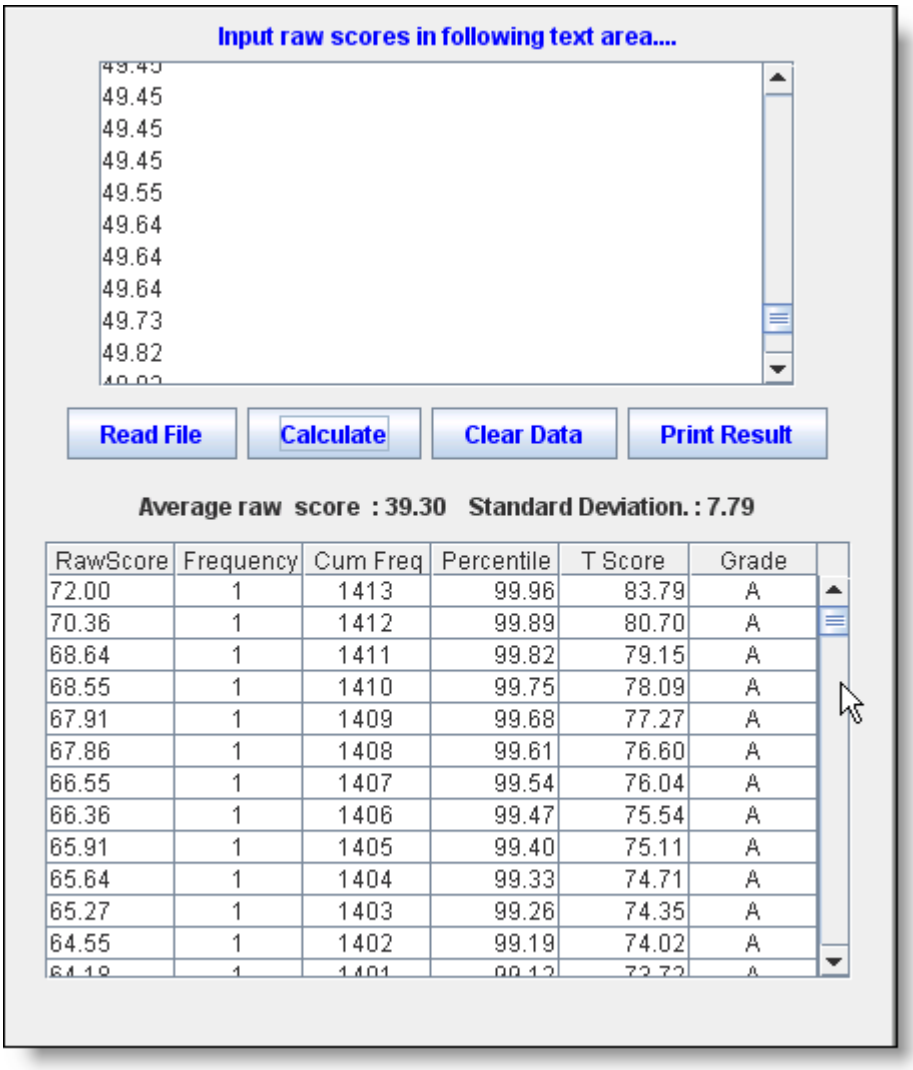
A	B	C	D	E	F	G	H
คะแนนดิบ	Tally	ความถี่	ความถี่สะสม	(cf+0.5f)	Area under Normal Curve	T	Grade
23	/	1	25	24.5	0.98	70.54	A
22	//	2	24	23	0.92	64.05	A
21	//	2	22	21	0.84	59.94	B
20	///	3	20	18.5	0.74	56.43	B
19	////	4	17	15	0.6	52.53	C
18	//	2	13	12	0.48	49.50	C
17	////	4	11	9	0.36	46.42	C
16	//	2	7	6	0.24	42.94	D
14	//	2	5	4	0.16	40.06	D
13	//	2	3	2	0.08	35.95	F
12	/	1	1	0.5	0.02	29.46	F

รูปที่ 4.12 ผลลัพธ์การคำนวณคะแนนทีปกติ ที่ได้จากโปรแกรมสำเร็จรูป Excel

การกำหนดระดับคะแนนหรือเกรด ยังคงใช้วิธีเดียวกันกับในการทดสอบ ในข้อ 1 ในตารางนี้เป็นการป้อนระดับคะแนนเข้าไปโดยผู้ใช้ มิได้เขียนเป็นคำสั่งให้โปรแกรม Excel คำนวณ

ข. ทดสอบกับข้อมูลจำนวน 1,413 คน เป็นผลการสอบวิชาฟิสิกส์ 1 ภาคการศึกษา 1/2550 สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี คะแนนดิบเก็บไว้ใน txt file ชื่อ P1_1_50.txt การตัดเกรดเลือกแบบ 5 เกรด คือ A, B, C, D และ F

ผลลัพธ์ที่ได้จากโปรแกรม TScoreCal ดังรูปที่ 4.13



รูป 4.13 แสดงผลลัพธ์ที่ได้จากโปรแกรม TScoreCal เมื่อข้อมูล 1413 คน

เมื่อนำข้อมูลทั้งหมด ไปหาความถี่ ความถี่สะสม เปอร์เซนไทล์ คะแนนทีปกติ และเขียนคำสั่ง VBA (Visual Basic Application) ให้กำหนดระดับคะแนนหรือเกรดในโปรแกรมสำเร็จรูป Excel พบว่าได้ผลลัพธ์ตรงกันทุกค่า โดยที่ เปอร์เซนไทล์ และ คะแนนทีปกติคิดที่เลขทศนิยม 2 ตำแหน่ง

ขั้นตอนการคำนวณหาค่าทางสถิติดังกล่าวในโปรแกรม Excel ได้แสดงไว้อย่างละเอียดในภาคผนวก B

บทที่ 5
สรุปวิจารณ์และข้อเสนอแนะ

จากการพัฒนาและทดสอบใช้งานโปรแกรมการคำนวณคะแนนทีปกติและกำหนดระดับคะแนน (Normalized T Score Calculation) หรือเรียกสั้น ๆ ว่า TScoreCal มีข้อสรุป ข้อสังเกต และข้อเสนอแนะในการใช้งานและการปรับปรุงในการพัฒนาครั้งต่อ ๆ ไปดังนี้

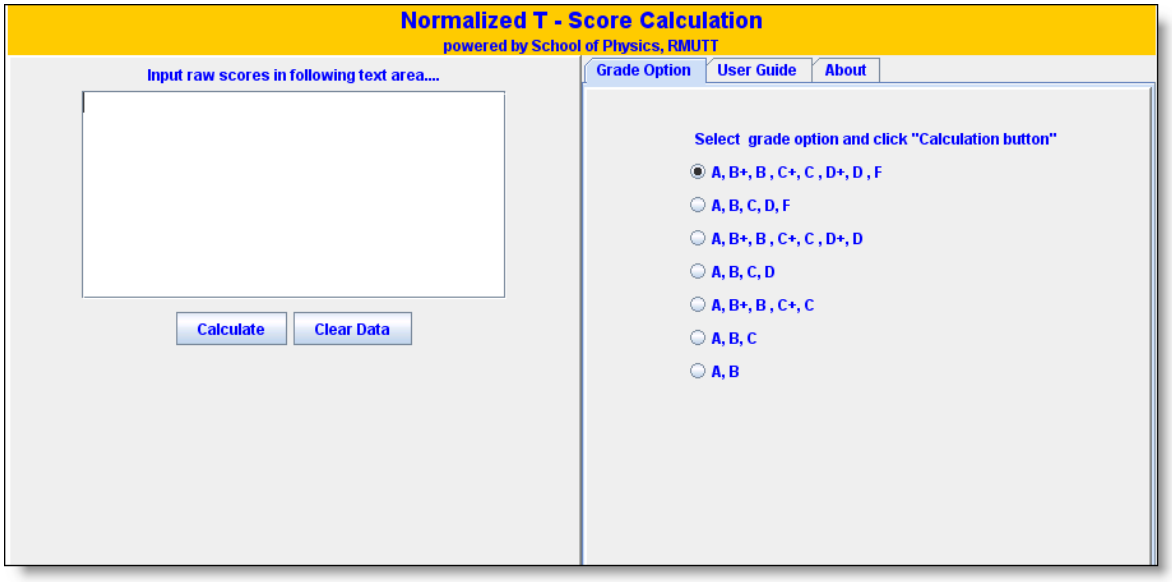
ในการออกแบบและพัฒนาโปรแกรมคำนวณคะแนนทีปกติและกำหนดระดับคะแนน **บรรลุผลสัมฤทธิ์ครบถ้วนตามวัตถุประสงค์ของการวิจัยทุกประการ** กล่าวคือ ผู้ใช้งานสามารถคำนวณคะแนนทีปกติและกำหนดระดับคะแนนโดยใช้งานผ่านเครือข่ายอินเตอร์เน็ตหรือ download โปรแกรมให้ทำงานในลักษณะ application ในเครื่องคอมพิวเตอร์ของตน ในการคำนวณคะแนนทีปกตินี้ ผู้ใช้งานไม่จำเป็นต้องใช้ตารางคะแนนทีเลย

เมื่อพิจารณาขอบเขตของการวิจัย พบว่า เป็นไปตามข้อกำหนดที่เขียนไว้ คือ

1. ประมวลผลจำนวนคะแนนนักศึกษาไม่เกิน 2,000 คน
2. ใช้ภาษาคอมพิวเตอร์ที่เรียกว่า จาวา(Java) ในการเขียนโปรแกรมทั้งหมด
3. สามารถทำงานผ่านเครือข่ายอินเตอร์เน็ตในลักษณะ applet
4. สามารถทำงานในลักษณะ application ซึ่งจะทำงานนอกบราวเซอร์ สามารถอ่านข้อมูลจากไฟล์ และสามารถส่งผลลัพธ์ที่ได้จากการคำนวณไปพิมพ์ที่เครื่องพิมพ์

มีข้อสังเกตในการพัฒนาโปรแกรมนี้ ดังต่อไปนี้

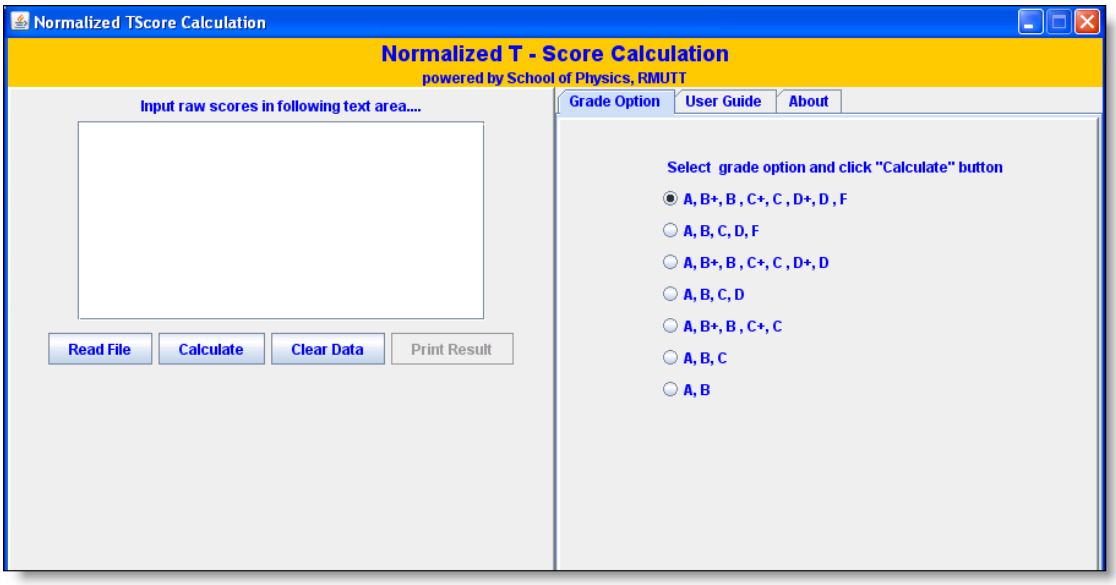
1. **การทำงานของโปรแกรมในรูปแบบ application จะมีฟังก์ชันมากกว่ารูปแบบแอพเพล็ต** หน้าตาของโปรแกรมที่ทำงานแบบแอพเพล็ต เป็นดังรูป 5.1



รูปที่ 5.1 โปรแกรม TScoreCal ทำงานแบบ applet ฝังตัวในบราวเซอร์

จะเห็นว่ามีปุ่มใต้ Text Area เพียง 2 ปุ่มคือ Calculate และ Clear Data ไม่มีปุ่ม Read file และ Print result ทั้งนี้เพราะข้อจำกัดในการทำงานของโปรแกรมแบบแอพเพล็ตที่ไม่ยอมให้มีการเขียนหรืออ่านไฟล์ หรือติดต่อกับพอร์ตใด ๆ (port) ในเครื่องคอมพิวเตอร์ของลูกข่าย จึงไม่สามารถส่งผลลัพธ์ออกไปพิมพ์ที่เครื่องพิมพ์ได้ ทั้งนี้เพื่อความปลอดภัยของเครื่องคอมพิวเตอร์ที่ดาวน์โหลดแอพเพล็ตมาใช้งาน

เมื่อเปลี่ยนรูปแบบการทำงานของโปรแกรมเป็นแบบ application ผู้ใช้งานเพียงแค่เชื่อมต่ออินเตอร์เน็ตไปยัง homepage ที่เก็บโปรแกรมนี้ คลิกลิงค์ที่ให้โปรแกรมนี้ทำงาน โปรแกรมจะถูกดาวน์โหลด มายังเครื่องคอมพิวเตอร์ของผู้ใช้งาน โดยอาศัย Java Network Launching Protocol โปรแกรมจะทำงานโดยที่ผู้ใช้งานไม่จำเป็นต้องทำการติดตั้งหรือ set up ทั้งนี้ในเครื่องคอมพิวเตอร์ผู้ใช้งานจะต้องมี JRE (Java Runtime Environment) ติดตั้งไว้เรียบร้อยแล้ว การทำงานของโปรแกรมในรูปแบบนี้ จะทำงานนอกบราวเซอร์ ผู้ใช้งานสามารถอ่านข้อมูลจากไฟล์และพิมพ์ผลลัพธ์ที่ได้จากการคำนวณเก็บไว้เป็นเอกสารได้

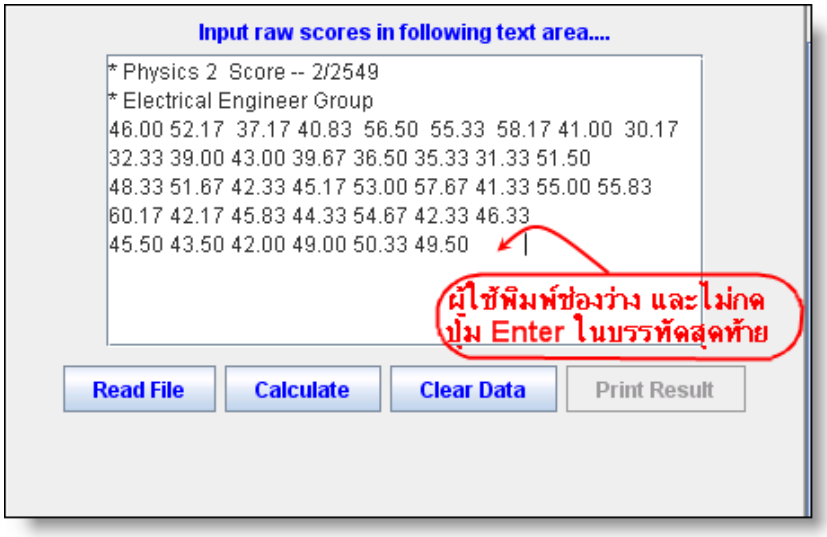


รูปที่ 5.2 โปรแกรมทำงานในรูปแบบ application มีปุ่ม Read file และ Print result เพิ่มขึ้นมา

ในการติดตั้งให้โปรแกรมทำงานในรูปแบบ application จะมีข้อยุ่งยากกว่ารูปแบบ applet เล็กน้อย คือ ต้องแก้ไขค่าระบบ(configure) ของ Web server ให้รู้จักไฟล์ ชนิด jnlp จะต้องบีบอัด class file ของโปรแกรมให้อยู่ในรูป jar file แล้วสร้างลายเซ็นกำกับลงใน jar file นี้ จะต้องสร้างเอกสาร jnlp เพื่อบอกให้ Java Web Start รู้คุณลักษณะของ jar file นี้ เพื่อกำหนดค่าให้โปรแกรมได้เริ่มต้นทำงานอย่างถูกต้อง (ดูรายละเอียดขั้นตอนการสร้างลายเซ็น TScoreCal.jar ในภาคผนวก C)

2. **ปัญหาปลีกย่อยในการป้อนคะแนนดิบลงในกรอบป้อนข้อความ**

- โปรแกรมรุ่นแรก ๆ พบว่าเมื่อป้อนคะแนนดิบในบรรทัดสุดท้าย และไม่ได้กดปุ่ม Enter ที่บรรทัดสุดท้าย คะแนนดิบค่าสุดท้ายจะไม่ถูกอ่านเข้าไปในหน่วยความจำ ได้แก้ไขข้อผิดพลาดเหล่านี้แล้วในรุ่นหลัง ๆ



รูปที่ 5.3 แสดงการป้อนคะแนนโดยไม่กดปุ่ม Enter ในบรรทัดสุดท้าย

- เมื่อพิมพ์คะแนนค่าสุดท้าย แล้วเคาะ spacebar และไม่กดปุ่ม Enter โปรแกรมจะเก็บค่า NULL ไว้ในตัวแปรอะเรย์เป็นคะแนนค่าสุดท้าย ทำให้ไม่สามารถเปลี่ยนคะแนนดิบให้เป็นจำนวนจริงได้ โปรแกรมจะหยุดการทำงาน (ถ้ารันโปรแกรมนี้ให้ทำงานเป็น desktop application จะเห็นการแจ้งการเกิด exception ซึ่งเป็นข้อผิดพลาดเกี่ยวกับการแปลงค่า NULL ให้เป็นจำนวนจริง) ผู้พัฒนาได้แก้ไขข้อผิดพลาดเหล่านี้แล้วในรุ่นหลัง ๆ เช่นกัน

- การพิมพ์คะแนนที่ติดลบ หรือใส่เครื่องหมายบวกหน้าคะแนน โปรแกรมจะแสดงข้อความผิดพลาด

3. **การคำนวณคะแนนทีปกติใช้ค่าความคลาดเคลื่อนที่ยอมรับได้** (Tolerance) **ที่** 10^{-7}

ในการพัฒนาในช่วงแรก ๆ ได้ตั้งค่าความคลาดเคลื่อนที่ยอมรับได้ไว้ที่ 10^{-5} พบว่าเมื่อข้อมูลมีเป็นจำนวนมาก (เกิน 1000 คนขึ้นไป) การคำนวณจะได้ค่าคะแนนทีปกติไม่ตรงกับค่าที่คำนวณได้จากโปรแกรมสำเร็จรูป Excel ณ ที่ทศนิยมตำแหน่งที่ 2 จะเกิดเหตุการณ์นี้ ตรงบริเวณที่คะแนนดิบมีค่าสูงมาก ๆ และต่ำมาก ๆ ส่วนคะแนนดิบที่มีค่ากลาง ๆ จะได้ผลลัพธ์ตรงกัน ผู้พัฒนาจึงขยายขอบเขตความคลาดเคลื่อนที่ยอมรับได้เป็น 10^{-7} ซึ่งจะได้ผลลัพธ์ของคะแนนทีปกติตรงกันกับค่าที่คำนวณได้จาก Excel ทุกค่า เมื่อคิดที่ทศนิยม 2 ตำแหน่ง

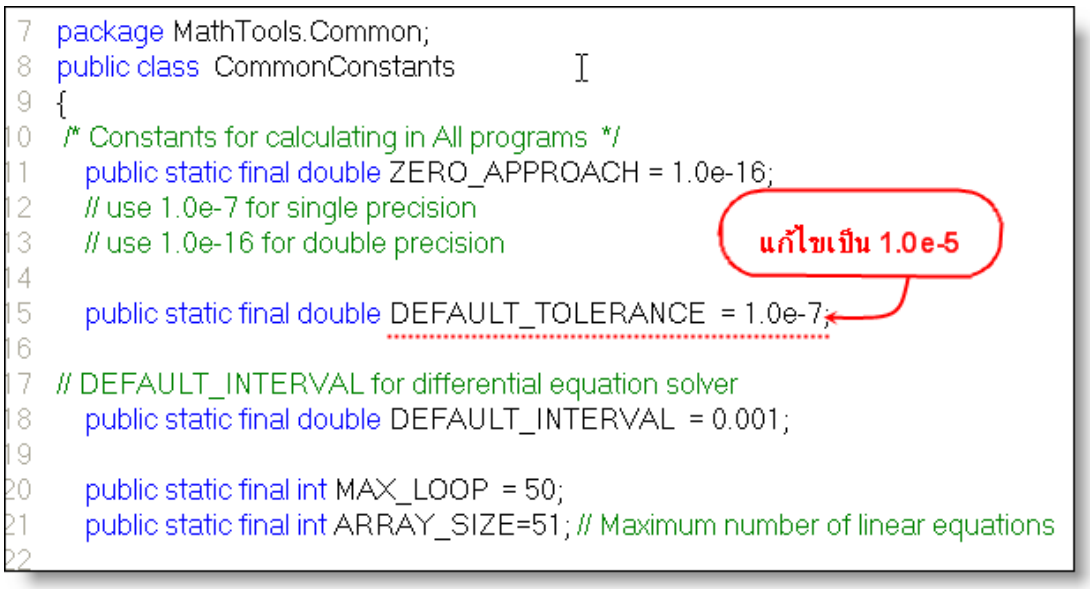
แต่ข้อเสียที่ตามมาคือโปรแกรมต้องใช้เวลาในการคำนวณมากขึ้น ถ้าข้อมูลมีเป็นจำนวนมาก ต้องทิ้งช่วงให้โปรแกรมคำนวณประมาณ 1-2 วินาทีผลลัพธ์จึงจะปรากฏขึ้นในตาราง ผู้พัฒนาได้แก้ไขปัญหานี้ โดยกำหนดพื้นที่ใต้เส้นโค้งปกติไว้เรียบร้อยแล้วเป็นบางค่า โปรแกรม

สามารถคำนวณพื้นที่ใต้โค้งปกติต่อไปได้เลย ไม่ต้องคำนวณคะแนนทีปกติจากค่าเริ่มต้นทุก ๆ ครั้ง source code ของคลาส SimpleStat ที่ได้รับการแก้ไขแล้ว ยกมาเฉพาะเมธอด findZAtKnownArea ( ) พิมพ์ให้เห็นชัดด้วยตัวหนา มีดังนี้

```
public double findZAtKnownArea (double area) throws
                                        SimpleStatException {
     boolean isAreaGreaterThanHalf = false;
     double lowerLimit=0;
     double newArea=0;
     double deltaX;
     double z=0 , sumArea=0;
     double dA;
        if ( area < 0 || area >1 ) throw new
SimpleStatException(SimpleStatException.BAD_AREA);
        if (area > 0.5){
            newArea = area - 0.5;
            isAreaGreaterThanHalf = true;
         } else {
            newArea = 0.5 - area;
        }

     if (newArea >= 0.01993880583837) {
           lowerLimit = 0.05;
           sumArea = 0.01993880583837;
        }
        if (newArea >= 0.09870632568292) {
           lowerLimit = 0.25;
           sumArea = 0.09870632568292;
        }
        if (newArea >= 0.19146246127401) {
           lowerLimit = 0.5;
           sumArea = 0.19146246127401;
        }
        if (newArea >= 0.34134474606854 ) {
           lowerLimit = 1.0;
           sumArea = 0.34134474606854;
        }
        if (newArea >= 0.43319279873114) {
           lowerLimit = 1.5;
           sumArea = 0.43319279873114;
        }
     if (newArea >= 0.47724986805182) {
           lowerLimit = 2;
           sumArea = 0.47724986805182;
        }

     if (newArea >= 0.48609655248650) {
           lowerLimit = 2.2;
           sumArea = 0.48609655248650;
        }
     if (newArea >= 0.48927588997832) {
           lowerLimit = 2.3;
           sumArea = 0.48927588997832;
        }
     if (newArea >= 0.49180246407540) {
           lowerLimit = 2.4;
           sumArea = 0.49180246407540;
```

```
        }
        if (newArea >= 0.49379033467422) {

           lowerLimit = 2.5;
           sumArea = 0.49379033467422;
        }
        if (newArea >=0.49653302619696) {
           lowerLimit = 2.7;
           sumArea = 0.49653302619696;
        }
        if (newArea >=0.49744486966957) {
           lowerLimit = 2.8;
           sumArea =0.49744486966957 ;
        }
        if (newArea >=0.49813418669962) {
           lowerLimit = 2.9;
           sumArea =0.49813418669962 ;
        }
        if (newArea >= 0.49865010196837) {
           lowerLimit = 3;
           sumArea = 0.49865010196837;
        }
     if (newArea >=0.49903239678678 ) {
           lowerLimit = 3.1;
           sumArea = 0.49903239678678;
        }
        if (newArea >= 0.49931286206208) {
           lowerLimit = 3.2;
           sumArea = 0.49931286206208;
        }
if (newArea >= 0.49951657585762) {
           lowerLimit = 3.3;
           sumArea = 0.49951657585762;
        }
        z = lowerLimit;
        deltaX = 0.00001;
        while( newArea - sumArea >DEFAULT_TOLERANCE) {
           dA = (1/sqrt(2*PI))*0.5*deltaX*(exp(-0.5*z*z) + exp(-
0.5*(z+deltaX)*(z+deltaX)));
           sumArea += dA;
           z += deltaX;
        }
     // System.out.println(" z in method = " + z);
     //       System.out.println(" Sumarea = " + sumArea);

        if (isAreaGreaterThanHalf)  return z+deltaX;
        else return -(z+deltaX);
     }
```

ถ้าการใช้งานต้องการคะแนนทีปกติที่ทศนิยมเพียง 1 ตำแหน่ง สามารถลดขอบเขตความคลาดเคลื่อนที่ยอมรับได้ให้เหลือเพียง 10^{-5} โดยแก้ไขค่า DEFAULT_TOLERANCE ในคลาส CommonConstants และลดค่า deltaX ให้เหลือเพียง 0.0001 ก็สามารถให้ผลลัพธ์ตามที่ต้องการได้ ทำให้โปรแกรมทำงานได้เร็วขึ้นด้วย




```
7  package MathTools.Common;
8  public class  CommonConstants
9  {
10  /* Constants for calculating in All programs  */
11    public static final double ZERO_APPROACH = 1.0e-16;
12    // use 1.0e-7 for single precision
13    // use 1.0e-16 for double precision
14
15    public static final double DEFAULT_TOLERANCE  = 1.0e-7;
16
17 // DEFAULT_INTERVAL for differential equation solver
18    public static final double DEFAULT_INTERVAL  = 0.001;
19
20    public static final int MAX_LOOP  = 50;
21    public static final int ARRAY_SIZE=51; // Maximum number of linear equations
22
```

รูปที่ 5.4 แสดงการลดค่า Tolerance ให้มีค่าน้อยลง

4. **การเรียงลำดับคะแนนดิบในโปรแกรมนี้ใช้การเรียงลำดับแบบ** Quick sort

การเรียงลำดับคะแนนดิบจากค่ามากไปหาค่าน้อยในโปรแกรมนี้ใช้การเรียงลำดับแบบ Quick sort โดยผู้พัฒนาได้นำ algorithm แบบ Quick sort มาออกแบบและเขียน source code ด้วยตนเอง ไม่ได้ใช้ ฟังก์ชัน sort ซึ่งเป็น API ของจาวาในคลาส Arraysแต่อย่างใด

รายละเอียดของเมธอด quicSort ( ) ของโปรแกรมนี้มีดังนี้

```
void quickSort(double[ ]item, int left, int right, Boolean ascending)
{
      int i,j;
      double comparand, temp;
                  i = left;
                  j = right;
                  comparand = item[(left+right)/2];
                  do { if (ascending) {
                        while( item[i] < comparand && i < right) i++;
                        while (comparand < item[j] && j > left) j--;
                  }else {
                        while( item[i] > comparand && i < right) i++;
                        while (comparand > item[j] && j > left) j--;
                  }
                  if ( i <= j) {
                        temp = item[i];
                        item[i] = item[j];
                        item[j] = temp;
                        i++;
                        j--;
                  }
               }while ( i <= j );
               if (left < j) quickSort(item, left, j, ascending);
               if(i < right ) quickSort(item, i,right, ascending);
               isSorted = true;
}
```

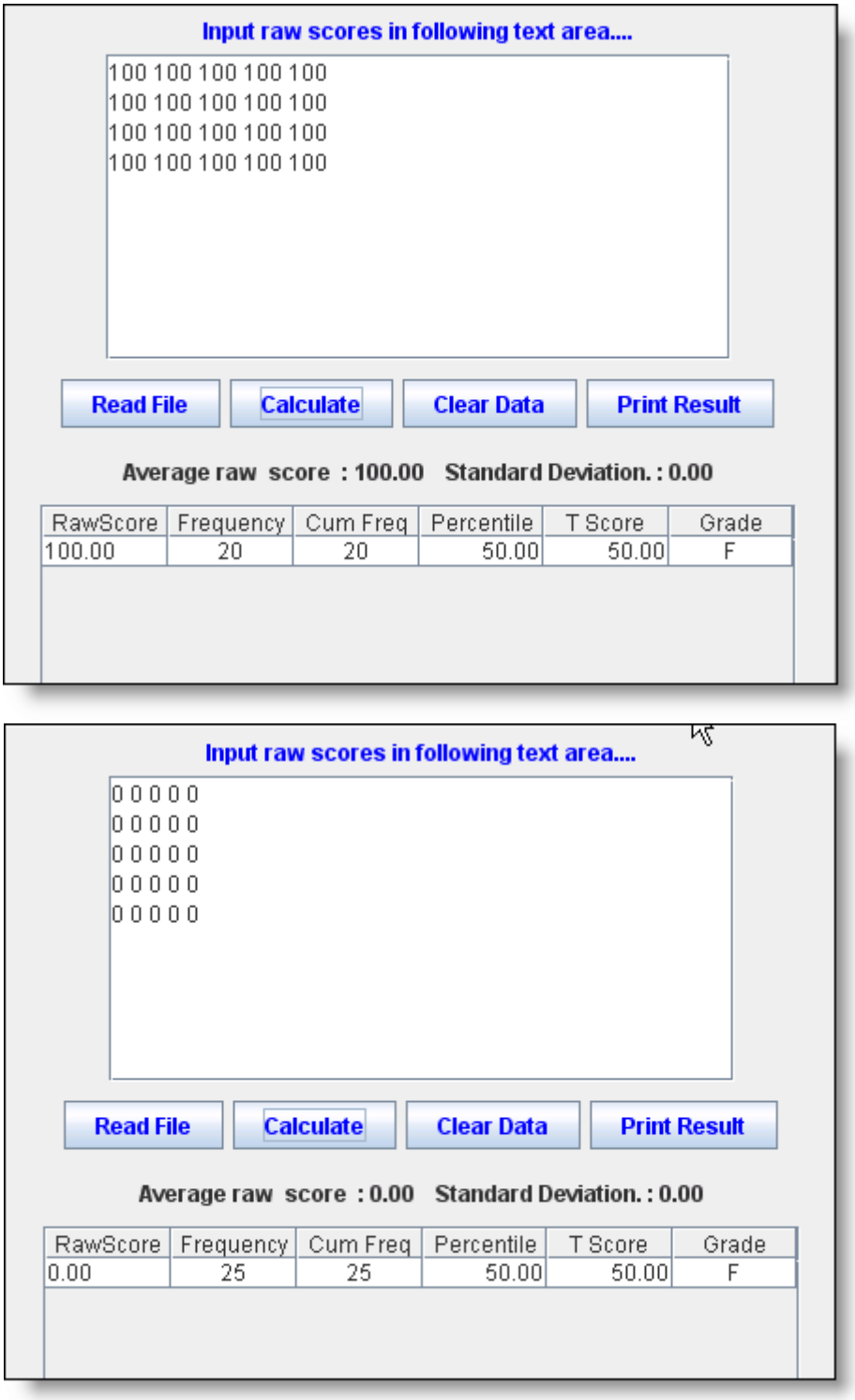
ได้ตรวจสอบการทำงานของฟังก์ชัน quickSort พบว่าเรียงลำดับคะแนนดิบได้อย่างถูกต้องและเป็นไปตามที่ต้องการ การเรียงลำดับจะเรียงจากค่าน้อยไปยังค่ามาก ถ้ากำหนดให้ Ascending มีค่าเป็นจริง และจะเรียงลำดับจากค่ามากไปยังค่าน้อย ถ้าตั้งค่า Ascending มีค่าเป็นเท็จ แต่ทั้งนี้ผู้พัฒนายังมิได้นำเมธอด quickSort นี้ไปเปรียบเทียบประสิทธิภาพการทำงาน กับวิธีการเรียงลำดับแบบอื่น ๆ เช่น Shell Sort insertion sort เป็นต้น

5. **การตัดเกรดที่แท้จริงควรใช้ดุลยพินิจของผู้สอนประกอบด้วย**

โปรแกรม TScoreCal ในรุ่นแรก ได้กำหนดช่วงของเกรด เป็น A, B, C, D และ F เท่านั้น ต่อมาทางมหาวิทยาลัยเทคโนโลยีราชมงคล ได้เพิ่มการตัดเกรดเป็นแบบ A B+ B C+ C D+ D และ F โปรแกรมนี้จึงได้เพิ่มทางเลือกให้ผู้ใช้งานหรือผู้สอนสามารถกำหนดช่วงเกรดได้มากขึ้น อย่างไรก็ตาม เกรดที่โปรแกรมนี้กำหนดไว้ให้เป็นแค่เพียงแนวทางสำหรับผู้สอนประกอบการตัดสินใจในการตัดเกรดเท่านั้น ผู้สอนอาจจะต้องคำนึงถึงองค์ประกอบอื่น ๆ เช่น ความตั้งใจเรียน ความขยันหมั่นเพียร และการตรงต่อเวลาของนักศึกษา ขอบเขตของเกรดแต่ละช่วงจึงอาจขยับขยายหรือเปลี่ยนแปลงได้ตามความเหมาะสมซึ่งขึ้นอยู่กับผู้สอนจะเห็นสมควร

ถ้านักศึกษาทุกคนทำคะแนนได้เท่ากันหมด ตัวอย่างเช่น ผลการสอบของนักศึกษา 20 คน ทุกคนได้คะแนน 100 คะแนนเท่ากันหมด (ซึ่งคงเป็นไปได้ยาก ในความเป็นจริง) นักศึกษาจะได้เกรดอะไร เมื่อทดลองป้อนให้ข้อมูลนี้คำนวณคะแนนทีปกติและตัดเกรด พบว่านักศึกษาทุกคนจะได้เกรด F !!

ทดลองเปลี่ยนคะแนนของนักศึกษาทุกคน จาก 100 คะแนน เป็นได้ 0 คะแนนทุกคน พบว่านักศึกษาทุกคนจะได้เกรด F เช่นเดียวกัน ดังรูปที่ 5.5



รูปที่ 5.5 ทดลองป้อนคะแนนที่เท่ากันจำนวน 20 คน

ข้อเสนอแนะสำหรับการพัฒนาเพิ่มขีดความสามารถของโปรแกรม

1. จุดประสงค์ของโปรแกรมนี้ต้องการให้ทำงานผ่านเครือข่ายอินเตอร์เน็ต จึงไม่ได้มีการออกแบบฐานข้อมูล เพื่อเก็บข้อมูลต่าง ๆ เช่น รหัสนักศึกษา ชื่อสกุล ชื่อรายวิชา คะแนนสอบกลางภาค และปลายภาค ซึ่งในระบบวัดผล – ประเมินผลนักศึกษา ควรมีข้อมูลเหล่านี้ ในการพัฒนาโปรแกรมให้ทำงานบน desktop computer หรือบนระบบเครือข่ายประจำถิ่น หรือ LAN (Local Area Network) ควรเพิ่มฐานข้อมูลเหล่านี้เข้าไปด้วย เพื่อความสมบูรณ์ในการใช้งาน

2. ควรเพิ่มฟังก์ชันในการ save ผลลัพธ์ที่ได้ในการคำนวณเพื่อเก็บไว้อ้างอิง หรือใช้งานร่วมกับโปรแกรมอื่นในภายหลังได้ โปรแกรม TScoreCal ที่พัฒนานี้สามารถเก็บผลลัพธ์โดยการพิมพ์ผ่านเครื่องพิมพ์เท่านั้น และกำหนดขนาดกระดาษไว้คงที่ที่ขนาด A4

ฟังก์ชัน Print result ของโปรแกรมนี้ ยังไม่มีความสามารถในการ preview ผลลัพธ์ของการพิมพ์ก่อนที่จะให้เครื่องพิมพ์พิมพ์ออกมาบนกระดาษ ในการพัฒนาครั้งต่อไปควรเพิ่มความสามารถในการ preview และให้ผู้ใช้งานสามารถเลือกชนิดและขนาดของกระดาษได้ตามความต้องการ

3. ควรเพิ่มขีดความสามารถในการกำหนดเกรด นอกจากโปรแกรมเป็นผู้กำหนดแล้ว ควรให้ ผู้ใช้สามารถกำหนดเกรดด้วยตนเองได้ด้วยว่าช่วงคะแนนทีปกติใดจึงจะได้ A, B, C, D หรือ F ผู้สอนบางท่านที่ทดลองใช้โปรแกรมนี้ ได้เรียกร้องให้ มีตัวเลือกช่วงการตัดเกรด เฉพาะ C, D และ F เพิ่มเข้าไปด้วย

ในการพัฒนาต่อยอดโปรแกรมนี้ให้มีขีดความสามารถเพิ่มขึ้น จึงควรสอบถามและรวบรวมความคิดเห็นของผู้ที่ได้ใช้โปรแกรมแล้ว เพื่อให้ได้โปรแกรมที่ตรงกับความต้องการของผู้ใช้งานจริง

บรรณานุกรม

1. ชวาล แพรัตกุล. 2526. **เทคนิคการวัดผล**. กรุงเทพฯ: ไทยวัฒนาพานิช.
2. ล้วน สายยศและอังคณา สายยศ. 2543. **เทคนิคการวัดผลการเรียนรู้**. พิมพ์ครั้งที่ 2. กรุงเทพฯ: สุวีริยาสาส์น.
3. ธานินทร์ ศิลปจารุ. 2550. การวิจัยและวิเคราะห์ข้อมูลทางสถิติด้วย SPSS. กรุงเทพฯ: วี. อินเตอร์ พริ้นท์.
4. พิชิต ฤทธิ์จรูญ. 2545. **หลักการวัดและประเมินผลการศึกษา**. พิมพ์ครั้งที่ 2. กรุงเทพฯ: เฮ้าส์ ออฟเดอร์มิสท์.
5. สมนึก ภัททิยธนี. 2546. **การวัดผลการศึกษา**. พิมพ์ครั้งที่ 4. กาฬสินธุ์: โรงพิมพ์ประสานการพิมพ์.
6. สมบูรณ์ ตันยะ. 2545. **การประเมินผลทางการศึกษา**. กรุงเทพฯ: สุวีริยาสาส์น.
7. สุพัฒน์ สุกมลสันต์. 2542. **การวิเคราะห์ข้อสอบและการตัดเกรดด้วยคอมพิวเตอร์**. กรุงเทพฯ: วิทยพัฒนา.
8. สุมาลี จันทร์ชลอ. 2542. **การวัดและประเมินผล**. กรุงเทพฯ: บริษัทพิมพ์ดีจำกัด.
9. Adamson, Chris and Marinacci, Joshua. 2005. **Swing Hacks**, New York: O'Reilly.
10. Bajpai, A.C. and others. 1990. **Engineering Mathematics**. 2nd ed. Singapore: John Willey & Sons, Inc.
11. Brian, cole. Eckstein, Robert. 2002. **Java Swing**. New York: O'Reilly.
12. Burden, Richard L. and Faires, Douglas J. 1993. **Numerical Analysis**. 5 th ed. Massachusetts: PWS Publishing
13. Cohen, Frank, 2004. **Java Testing and Design**, New York: Prentice-Hall.
14. Deitel, H.M. and others. 2001, **Advanced Java How to Program**, New York: Prentice-Hall.
15. Dejong, Marvin L. 1991. **Introduction to Computational Physics**. New York: Addison -Wesley.
16. Garrido, Jose, M. 2003. **Object Oriented Programming from Problem Solving to Java**. Charles River Media.
17. Gutz, Steven. 2000, **Up to Speed with Swing**, 2nd ed. Greenwich: Manning Publication.

18. Henrici,P. 1964. **Elements of Numerical Analysis**. New York: John Willey & Sons, Inc.

19. Kreyszig, Erwin. 1988, **Advance Engineering Mathematics**. New York: John Willey & Sons, Inc.

20. O'Hagen, Don. 2005. **Core Java 2, Vol II.** Sun Micro System, Inc.

21. Palmer, Grunt.. 2003. **Technical Java: Developing Scientific and Engineering Application**. U.S.A.: Pearson Education, Inc.

22. Pantham, Satyaraj. 1999. **Pure JFC Swing.** U.S.A.: Sam Publishing.

23. Richardson, W. Clay. 2007. **Professional Java 6**. U.S.A.: Wiley Publishing.

24. Simmons, Robert. 2004. **Hardcore Java.** New York: O'Reilly.

25. Topley, Kim. 1999. **Core Swing Advanced Programming**. New Jersey: Prentice Hall.

26. Walrath, Kathy and others. 2004. **JFC Swing Tutorial: A Guide to Constructing GUIs.** 2 nd ed, California: Addison Wesley.

ภาคผนวก A
ผลการคำนวณหาพื้นที่ใต้เส้นโค้งปกติ และค่า Z เปรียบเทียบกับผลลัพธ์จากการใช้ฟังก์ชันที่มีอยู่ใน Excel

ตารางต่อไปนี้แสดงผลการคำนวณหาพื้นที่ใต้เส้นโค้งปกติในช่วงค่า Z ตั้งแต่ -1.90 จนถึง 1.75 โดยเพิ่มค่า Z ครั้งละ 0.01

พื้นที่ใต้เส้นโค้งปกติที่คำนวณได้ เป็นการเรียกใช้เมธอดfindAreaUnderNormalCurve ( ) ที่มีอยู่ในคลาส SImpleStat คำนวณ เป็นการอินทิเกรตหาพื้นที่ใต้โค้งปกติซึ่งมีรูปสมการ

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$$

$$\text{พื้นที่ใต้เส้นโค้งปกติ} = \int_{lower_z}^{upper_z} f(z)dz$$

โดยใช้วิธีของ Simpson 1/3 โดยมีขีดจำกัดล่างที่ lower_z และขีดจำกัดบนที่ upper_z

```
public double findAreaUnderNormalCurve(double lower_z,double upper_z) {

            Function normalFunction = new Function() {
      public double Of(double x) {
            return (1/sqrt(2.0*PI)*exp(-(x*x)/2.0));}
      };
            Simpson1_3Integration si = new Simpson1_3Integration
                  (normalFunction,lower_z, upper_z,1000);
            return (si.getIntegrationResult());
}
```

พื้นที่ใต้เส้นโค้งปกติที่ได้จากโปรแกรมสำเร็จรูป Excel เป็นการใช้ฟังก์ชันที่มีอยู่ในตัวมันเอง คือ NORMSDIST คำนวณ

ตาราง A1 เปรียบเทียบพื้นที่ใต้เส้นโค้งปกติที่คำนวณได้จากโปรแกรม TScoreCal กับฟังก์ชัน NormSDist ของ Excel

Z value	Area under Normal Curve From Program	Area Under Normal Curve From Function NORMSDIST(cell) in Excel	Z value	Area under Normal Curve From Program	Area Under Normal Curve From Function NORMSDIST(cell) in Excel
-1.90	0.0287	0.0287	-1.42	0.0778	0.0778
-1.89	0.0294	0.0294	-1.41	0.0793	0.0793
-1.88	0.0301	0.0301	-1.40	0.0808	0.0808
-1.87	0.0307	0.0307	-1.39	0.0823	0.0823
-1.86	0.0314	0.0314	-1.38	0.0838	0.0838
-1.85	0.0322	0.0322	-1.37	0.0853	0.0853
-1.84	0.0329	0.0329	-1.36	0.0869	0.0869
-1.83	0.0336	0.0336	-1.35	0.0885	0.0885
-1.82	0.0344	0.0344	-1.34	0.0901	0.0901
-1.81	0.0351	0.0351	-1.33	0.0918	0.0918
-1.80	0.0359	0.0359	-1.32	0.0934	0.0934
-1.79	0.0367	0.0367	-1.31	0.0951	0.0951
-1.78	0.0375	0.0375	-1.30	0.0968	0.0968
-1.77	0.0384	0.0384	-1.29	0.0985	0.0985
-1.76	0.0392	0.0392	-1.28	0.1003	0.1003
-1.75	0.0401	0.0401	-1.27	0.1020	0.1020
-1.74	0.0409	0.0409	-1.26	0.1038	0.1038
-1.73	0.0418	0.0418	-1.25	0.1056	0.1056
-1.72	0.0427	0.0427	-1.24	0.1075	0.1075
-1.71	0.0436	0.0436	-1.23	0.1093	0.1093
-1.70	0.0446	0.0446	-1.22	0.1112	0.1112
-1.69	0.0455	0.0455	-1.21	0.1131	0.1131
-1.68	0.0465	0.0465	-1.20	0.1151	0.1151
-1.67	0.0475	0.0475	-1.19	0.1170	0.1170
-1.66	0.0485	0.0485	-1.18	0.1190	0.1190
-1.65	0.0495	0.0495	-1.17	0.1210	0.1210
-1.64	0.0505	0.0505	-1.16	0.1230	0.1230
-1.63	0.0516	0.0516	-1.15	0.1251	0.1251
-1.62	0.0526	0.0526	-1.14	0.1271	0.1271
-1.61	0.0537	0.0537	-1.13	0.1292	0.1292
-1.60	0.0548	0.0548	-1.12	0.1314	0.1314
-1.59	0.0559	0.0559	-1.11	0.1335	0.1335
-1.58	0.0571	0.0571	-1.10	0.1357	0.1357
-1.57	0.0582	0.0582	-1.09	0.1379	0.1379
-1.56	0.0594	0.0594	-1.08	0.1401	0.1401
-1.55	0.0606	0.0606	-1.07	0.1423	0.1423
-1.54	0.0618	0.0618	-1.06	0.1446	0.1446
-1.53	0.0630	0.0630	-1.05	0.1469	0.1469
-1.52	0.0643	0.0643	-1.04	0.1492	0.1492
-1.51	0.0655	0.0655	-1.03	0.1515	0.1515
-1.50	0.0668	0.0668	-1.02	0.1539	0.1539
-1.49	0.0681	0.0681	-1.01	0.1562	0.1562
-1.48	0.0694	0.0694	-1.00	0.1587	0.1587
-1.47	0.0708	0.0708	-0.99	0.1611	0.1611
-1.46	0.0721	0.0721	-0.98	0.1635	0.1635
-1.45	0.0735	0.0735	-0.97	0.1660	0.1660
-1.44	0.0749	0.0749	-0.96	0.1685	0.1685
-1.43	0.0764	0.0764	-0.95	0.1711	0.1711

Z value	Area under Normal Curve From Program	Area Under Normal Curve From Function NORMSDIST(cell) in Excel	Z value	Area under Normal Curve From Program	Area Under Normal Curve From Function NORMSDIST(cell) in Excel
-0.94	0.1736	0.1736	-0.49	0.3121	0.3121
-0.93	0.1762	0.1762	-0.48	0.3156	0.3156
-0.92	0.1788	0.1788	-0.47	0.3192	0.3192
-0.91	0.1814	0.1814	-0.46	0.3228	0.3228
-0.90	0.1841	0.1841	-0.45	0.3264	0.3264
-0.89	0.1867	0.1867	-0.44	0.3300	0.3300
-0.88	0.1894	0.1894	-0.43	0.3336	0.3336
-0.87	0.1922	0.1922	-0.42	0.3372	0.3372
-0.86	0.1949	0.1949	-0.41	0.3409	0.3409
-0.85	0.1977	0.1977	-0.40	0.3446	0.3446
-0.84	0.2005	0.2005	-0.39	0.3483	0.3483
-0.83	0.2033	0.2033	-0.38	0.3520	0.3520
-0.82	0.2061	0.2061	-0.37	0.3557	0.3557
-0.81	0.2090	0.2090	-0.36	0.3594	0.3594
-0.80	0.2119	0.2119	-0.35	0.3632	0.3632
-0.79	0.2148	0.2148	-0.34	0.3669	0.3669
-0.78	0.2177	0.2177	-0.33	0.3707	0.3707
-0.77	0.2206	0.2206	-0.32	0.3745	0.3745
-0.76	0.2236	0.2236	-0.31	0.3783	0.3783
-0.75	0.2266	0.2266	-0.30	0.3821	0.3821
-0.74	0.2296	0.2296	-0.29	0.3859	0.3859
-0.73	0.2327	0.2327	-0.28	0.3897	0.3897
-0.72	0.2358	0.2358	-0.27	0.3936	0.3936
-0.71	0.2389	0.2389	-0.26	0.3974	0.3974
-0.70	0.2420	0.2420	-0.25	0.4013	0.4013
-0.69	0.2451	0.2451	-0.24	0.4052	0.4052
-0.68	0.2483	0.2483	-0.23	0.4090	0.4090
-0.67	0.2514	0.2514	-0.22	0.4129	0.4129
-0.66	0.2546	0.2546	-0.21	0.4168	0.4168
-0.65	0.2578	0.2578	-0.20	0.4207	0.4207
-0.64	0.2611	0.2611	-0.19	0.4247	0.4247
-0.63	0.2643	0.2643	-0.18	0.4286	0.4286
-0.62	0.2676	0.2676	-0.17	0.4325	0.4325
-0.61	0.2709	0.2709	-0.16	0.4364	0.4364
-0.60	0.2743	0.2743	-0.15	0.4404	0.4404
-0.59	0.2776	0.2776	-0.14	0.4443	0.4443
-0.58	0.2810	0.2810	-0.13	0.4483	0.4483
-0.57	0.2843	0.2843	-0.12	0.4522	0.4522
-0.56	0.2877	0.2877	-0.11	0.4562	0.4562
-0.55	0.2912	0.2912	-0.10	0.4602	0.4602
-0.54	0.2946	0.2946	-0.09	0.4641	0.4641
-0.53	0.2981	0.2981	-0.08	0.4681	0.4681
-0.52	0.3015	0.3015	-0.07	0.4721	0.4721
-0.51	0.3050	0.3050	-0.06	0.4761	0.4761
-0.50	0.3085	0.3085	-0.05	0.4801	0.4801

Z value	Area under Normal Curve From Program	Area Under Normal Curve From Function NORMSDIST(cell) in Excel	Z value	Area under Normal Curve From Program	Area Under Normal Curve From Function NORMSDIST(cell) in Excel
-0.04	0.4840	0.4840	0.41	0.6591	0.6591
-0.03	0.4880	0.4880	0.42	0.6628	0.6628
-0.02	0.4920	0.4920	0.43	0.6664	0.6664
-0.01	0.4960	0.4960	0.44	0.6700	0.6700
0.00	0.5000	0.5000	0.45	0.6736	0.6736
0.01	0.5040	0.5040	0.46	0.6772	0.6772
0.02	0.5080	0.5080	0.47	0.6808	0.6808
0.03	0.5120	0.5120	0.48	0.6844	0.6844
0.04	0.5160	0.5160	0.49	0.6879	0.6879
0.05	0.5199	0.5199	0.50	0.6915	0.6915
0.06	0.5239	0.5239	0.51	0.6950	0.6950
0.07	0.5279	0.5279	0.52	0.6985	0.6985
0.08	0.5319	0.5319	0.53	0.7019	0.7019
0.09	0.5359	0.5359	0.54	0.7054	0.7054
0.10	0.5398	0.5398	0.55	0.7088	0.7088
0.11	0.5438	0.5438	0.56	0.7123	0.7123
0.12	0.5478	0.5478	0.57	0.7157	0.7157
0.13	0.5517	0.5517	0.58	0.7190	0.7190
0.14	0.5557	0.5557	0.59	0.7224	0.7224
0.15	0.5596	0.5596	0.60	0.7257	0.7257
0.16	0.5636	0.5636	0.61	0.7291	0.7291
0.17	0.5675	0.5675	0.62	0.7324	0.7324
0.18	0.5714	0.5714	0.63	0.7357	0.7357
0.19	0.5753	0.5753	0.64	0.7389	0.7389
0.20	0.5793	0.5793	0.65	0.7422	0.7422
0.21	0.5832	0.5832	0.66	0.7454	0.7454
0.22	0.5871	0.5871	0.67	0.7486	0.7486
0.23	0.5910	0.5910	0.68	0.7517	0.7517
0.24	0.5948	0.5948	0.69	0.7549	0.7549
0.25	0.5987	0.5987	0.70	0.7580	0.7580
0.26	0.6026	0.6026	0.71	0.7611	0.7611
0.27	0.6064	0.6064	0.72	0.7642	0.7642
0.28	0.6103	0.6103	0.73	0.7673	0.7673
0.29	0.6141	0.6141	0.74	0.7704	0.7704
0.30	0.6179	0.6179	0.75	0.7734	0.7734
0.31	0.6217	0.6217	0.76	0.7764	0.7764
0.32	0.6255	0.6255	0.77	0.7794	0.7794
0.33	0.6293	0.6293	0.78	0.7823	0.7823
0.34	0.6331	0.6331	0.79	0.7852	0.7852
0.35	0.6368	0.6368	0.80	0.7881	0.7881
0.36	0.6406	0.6406	0.81	0.7910	0.7910
0.37	0.6443	0.6443	0.82	0.7939	0.7939
0.38	0.6480	0.6480	0.83	0.7967	0.7967
0.39	0.6517	0.6517	0.84	0.7995	0.7995
0.40	0.6554	0.6554	0.85	0.8023	0.8023

Z value	Area under Normal Curve From Program	Area Under Normal Curve From Function NORMSDIST(cell) in Excel	Z value	Area under Normal Curve From Program	Area Under Normal Curve From Function NORMSDIST(cell) in Excel
0.86	0.8051	0.8051	1.31	0.9049	0.9049
0.87	0.8078	0.8078	1.32	0.9066	0.9066
0.88	0.8106	0.8106	1.33	0.9082	0.9082
0.89	0.8133	0.8133	1.34	0.9099	0.9099
0.90	0.8159	0.8159	1.35	0.9115	0.9115
0.91	0.8186	0.8186	1.36	0.9131	0.9131
0.92	0.8212	0.8212	1.37	0.9147	0.9147
0.93	0.8238	0.8238	1.38	0.9162	0.9162
0.94	0.8264	0.8264	1.39	0.9177	0.9177
0.95	0.8289	0.8289	1.40	0.9192	0.9192
0.96	0.8315	0.8315	1.41	0.9207	0.9207
0.97	0.8340	0.8340	1.42	0.9222	0.9222
0.98	0.8365	0.8365	1.43	0.9236	0.9236
0.99	0.8389	0.8389	1.44	0.9251	0.9251
1.00	0.8413	0.8413	1.45	0.9265	0.9265
1.01	0.8438	0.8438	1.46	0.9279	0.9279
1.02	0.8461	0.8461	1.47	0.9292	0.9292
1.03	0.8485	0.8485	1.48	0.9306	0.9306
1.04	0.8508	0.8508	1.49	0.9319	0.9319
1.05	0.8531	0.8531	1.50	0.9332	0.9332
1.06	0.8554	0.8554	1.51	0.9345	0.9345
1.07	0.8577	0.8577	1.52	0.9357	0.9357
1.08	0.8599	0.8599	1.53	0.9370	0.9370
1.09	0.8621	0.8621	1.54	0.9382	0.9382
1.10	0.8643	0.8643	1.55	0.9394	0.9394
1.11	0.8665	0.8665	1.56	0.9406	0.9406
1.12	0.8686	0.8686	1.57	0.9418	0.9418
1.13	0.8708	0.8708	1.58	0.9429	0.9429
1.14	0.8729	0.8729	1.59	0.9441	0.9441
1.15	0.8749	0.8749	1.60	0.9452	0.9452
1.16	0.8770	0.8770	1.61	0.9463	0.9463
1.17	0.8790	0.8790	1.62	0.9474	0.9474
1.18	0.8810	0.8810	1.63	0.9484	0.9484
1.19	0.8830	0.8830	1.64	0.9495	0.9495
1.20	0.8849	0.8849	1.65	0.9505	0.9505
1.21	0.8869	0.8869	1.66	0.9515	0.9515
1.22	0.8888	0.8888	1.67	0.9525	0.9525
1.23	0.8907	0.8907	1.68	0.9535	0.9535
1.24	0.8925	0.8925	1.69	0.9545	0.9545
1.25	0.8944	0.8944	1.70	0.9554	0.9554
1.26	0.8962	0.8962	1.71	0.9564	0.9564
1.27	0.8980	0.8980	1.72	0.9573	0.9573
1.28	0.8997	0.8997	1.73	0.9582	0.9582
1.29	0.9015	0.9015	1.74	0.9591	0.9591
1.30	0.9032	0.9032	1.75	0.9599	0.9599

ในทางกลับกัน เมื่อรู้พื้นที่ใต้โค้งปกติ จะคำนวณหาค่า Z ที่ทำให้เกิดพื้นที่ใต้โค้งค่าเท่านี้ เมธอดที่ทำหน้าที่คำนวณ คือ findZAtKnownArea (area) ซึ่งอยู่ในคลาส SimpleStat เช่นกัน การคำนวณจะแบ่งพื้นที่ออกเป็นสี่เหลี่ยมคางหมูเล็ก ๆ เป็นจำนวน รวมพื้นที่สี่เหลี่ยมคางหมู โดยเริ่มจากทางซ้ายมือ สะสมพื้นที่ไปเรื่อย ๆ จนถึงค่าของพื้นที่ที่กำหนดให้ ก็จะสามารถรู้ค่า Z ที่ทำให้เกิดพื้นที่ใต้เส้นโค้งปกติค่านี้ได้ รายละเอียดเฉพาะเมธอด findZAtKnownArea มีดังนี้

```
/** finding statistic Z value at known area under normal curve
* @param area under normal curve should be between 0 to 1.
* @return z value at known area
*/
public double findZAtKnownArea (double area)
       throws SimpleStatException {
      boolean isAreaGreaterThanHalf = false;
      double lowerLimit=0;
      double newArea=0;
      double deltaX;
      double z=0 , sumArea=0;
      double dA;
            if ( area < 0 || area >1 ) throw
      new SimpleStatException(SimpleStatException.BAD_AREA);
            if (area > 0.5){
                  newArea = area - 0.5;
                  isAreaGreaterThanHalf = true;
       } else {
                  newArea = 0.5 - area;
            }
      if (newArea >= 0.01993880583837) {
                  lowerLimit = 0.05;
                  sumArea = 0.01993880583837;
            }
      if (newArea >= 0.09870632568292) {
                  lowerLimit = 0.25;
                  sumArea = 0.09870632568292;
            }
      if (newArea >= 0.19146246127401) {
                  lowerLimit = 0.5;
                  sumArea = 0.19146246127401;
            }
      if (newArea >= 0.34134474606854 ) {
                  lowerLimit = 1.0;
                  sumArea = 0.34134474606854;
            }
      if (newArea >= 0.43319279873114) {
                  lowerLimit = 1.5;
                  sumArea = 0.43319279873114;
            }
      if (newArea >= 0.47724986805182) {
                  lowerLimit = 2;
                  sumArea = 0.47724986805182;
            }
      if (newArea >= 0.49379033467422) {
                  lowerLimit = 2.5;
                  sumArea = 0.49379033467422;
            }
      if (newArea >= 0.49865010196837) {
                  lowerLimit = 3;
```

```
                        sumArea = 0.49865010196837;
                }
        z = lowerLimit;
        deltaX = 0.00001;
        while( newArea - sumArea >0.00001) {
                        dA = (1/sqrt(2*PI))*0.5*deltaX*(exp(-0.5*z*z)
                        + exp(-0.5*(z+deltaX)*(z+deltaX)));
                        sumArea += dA;
                        z += deltaX;
        }

              if (isAreaGreaterThanHalf)  return z+deltaX;
              else return -(z+deltaX);
    }
```

การทดสอบผลการคำนวณหาค่า Z เมื่อทราบพื้นที่ใต้เส้นโค้งปกติ ทำได้โดยสร้างคลาสชื่อ InverseZArea มีรายละเอียดดังนี้

```
/* File :InverseZArea.java
* Project: Math Tools in Java
* Purpose: Test class for finding Z corresponding to specified area.
* Author : Wachara R.
* First Released: :  Mon 7 July 2008
*/
import static java.lang.Math.*;
import java.io.*;
import java.util.*;
import MathTools.Common.*;
import MathTools.SimpleStat.*;

public class InverseZArea{
public static void main(String[] args) {
double z;
double [ ] area = {
0.0287,0.0294,0.0301,0.0307,0.0314,0.0322,0.0329,0.0336,0.0344,0.0351,
0.0359,0.0367,0.0375,0.0384,0.0392,0.0401,0.0409,0.0418,0.0427,0.0436,
0.0446,0.0455,0.0465,0.0475,0.0485,0.0495,0.0505,0.0516,0.0526,0.0537,
0.0548,0.0559,0.0571,0.0582,0.0594,0.0606,0.0618,0.0630,0.0643,0.0655,
0.0668,0.0681,0.0694,0.0708,0.0721,0.0735,0.0749,0.0764,0.0778,0.0793,
0.0808,0.0823,0.0838,0.0853,0.0869,0.0885,0.0901,0.0918,0.0934,0.0951,
0.0968,0.0985,0.1003,0.1020,0.1038,0.1056,0.1075,0.1093,0.1112,0.1131,
0.1151,0.1170,0.1190,0.1210,0.1230,0.1251,0.1271,0.1292,0.1314,0.1335,
0.1357,0.1379,0.1401,0.1423,0.1446,0.1469,0.1492,0.1515,0.1539,0.1562,
0.1587,0.1611,0.1635,0.1660,0.1685,0.1711,0.1736,0.1762,0.1788,0.1814,
0.1841,0.1867,0.1894,0.1922,0.1949,0.1977,0.2005,0.2033,0.2061,0.2090,
0.2119,0.2148,0.2177,0.2206,0.2236,0.2266,0.2296,0.2327,0.2358,0.2389,
0.2420,0.2451,0.2483,0.2514,0.2546,0.2578,0.2611,0.2643,0.2676,0.2709,
0.2743,0.2776,0.2810,0.2843,0.2877,0.2912,0.2946,0.2981,0.3015,0.3050,
0.3085,0.3121,0.3156,0.3192,0.3228,0.3264,0.3300,0.3336,0.3372,0.3409,
0.3446,0.3483,0.3520,0.3557,0.3594,0.3632,0.3669,0.3707,0.3745,0.3783,
0.3821,0.3859,0.3897,0.3936,0.3974,0.4013,0.4052,0.4090,0.4129,0.4168,
```

```
0.4207,0.4247,0.4286,0.4325,0.4364,0.4404,0.4443,0.4483,0.4522,0.4562,
0.4602,0.4641,0.4681,0.4721,0.4761,0.4801,0.4840,0.4880,0.4920,0.4960,
0.5000,0.5040,0.5080,0.5120,0.5160,0.5199,0.5239,0.5279,0.5319,0.5359,
0.5398,0.5438,0.5478,0.5517,0.5557,0.5596,0.5636,0.5675,0.5714,0.5753,
0.5793,0.5832,0.5871,0.5910,0.5948,0.5987,0.6026,0.6064,0.6103,0.6141,
0.6179,0.6217,0.6255,0.6293,0.6331,0.6368,0.6406,0.6443,0.6480,0.6517,
0.6554,0.6591,0.6628,0.6664,0.6700,0.6736,0.6772,0.6808,0.6844,0.6879,
0.6915,0.6950,0.6985,0.7019,0.7054,0.7088,0.7123,0.7157,0.7190,0.7224,
0.7257,0.7291,0.7324,0.7357,0.7389,0.7422,0.7454,0.7486,0.7517,0.7549,
0.7580,0.7611,0.7642,0.7673,0.7704,0.7734,0.7764,0.7794,0.7823,0.7852,
0.7881,0.7910,0.7939,0.7967,0.7995,0.8023,0.8051,0.8078,0.8106,0.8133,
0.8159,0.8186,0.8212,0.8238,0.8264,0.8289,0.8315,0.8340,0.8365,0.8389,
0.8413,0.8438,0.8461,0.8485,0.8508,0.8531,0.8554,0.8577,0.8599,0.8621,
0.8643,0.8665,0.8686,0.8708,0.8729,0.8749,0.8770,0.8790,0.8810,0.8830,
0.8849,0.8869,0.8888,0.8907,0.8925,0.8944,0.8962,0.8980,0.8997,0.9015,
0.9032,0.9049,0.9066,0.9082,0.9099,0.9115,0.9131,0.9147,0.9162,0.9177,
};
int i;
      SimpleStat s = new SimpleStat();
            try  {
            for( i = 0; i < area.length ; i++) {
                  z = s.findZAtKnownArea(area[i]);
                  System.out.printf("%4.4f    \t%3.2f\n",    area[i],z);

            }
            } catch (Exception err ) { System.out.println(err);   }
} // main
}
```

ตาราง A2 เปรียบเทียบค่าที่คำนวณได้จากโปรแกรม และผลลัพธ์ที่ได้จากการใช้คำสั่ง NormSInv ซึ่งเป็นฟังก์ชัน Built in ใน Excel

Area Under Normal curve	Value of Z From Program	Value of Z From Excel	Area Under Normal curve	Value of Z From Program	Value of Z From Excel
0.0287	-1.90	-1.90	0.0778	-1.42	-1.42
0.0294	-1.89	-1.89	0.0793	-1.41	-1.41
0.0301	-1.88	-1.88	0.0808	-1.40	-1.40
0.0307	-1.87	-1.87	0.0823	-1.39	-1.39
0.0314	-1.86	-1.86	0.0838	-1.38	-1.38
0.0322	-1.85	-1.85	0.0853	-1.37	-1.37
0.0329	-1.84	-1.84	0.0869	-1.36	-1.36
0.0336	-1.83	-1.83	0.0885	-1.35	-1.35
0.0344	-1.82	-1.82	0.0901	-1.34	-1.34
0.0351	-1.81	-1.81	0.0918	-1.33	-1.33
0.0359	-1.80	-1.80	0.0934	-1.32	-1.32
0.0367	-1.79	-1.79	0.0951	-1.31	-1.31
0.0375	-1.78	-1.78	0.0968	-1.30	-1.30
0.0384	-1.77	-1.77	0.0985	-1.29	-1.29
0.0392	-1.76	-1.76	0.1003	-1.28	-1.28
0.0401	-1.75	-1.75	0.1020	-1.27	-1.27
0.0409	-1.74	-1.74	0.1038	-1.26	-1.26
0.0418	-1.73	-1.73	0.1056	-1.25	-1.25
0.0427	-1.72	-1.72	0.1075	-1.24	-1.24
0.0436	-1.71	-1.71	0.1093	-1.23	-1.23
0.0446	-1.70	-1.70	0.1112	-1.22	-1.22
0.0455	-1.69	-1.69	0.1131	-1.21	-1.21
0.0465	-1.68	-1.68	0.1151	-1.20	-1.20
0.0475	-1.67	-1.67	0.1170	-1.19	-1.19
0.0485	-1.66	-1.66	0.1190	-1.18	-1.18
0.0495	-1.65	-1.65	0.1210	-1.17	-1.17
0.0505	-1.64	-1.64	0.1230	-1.16	-1.16
0.0516	-1.63	-1.63	0.1251	-1.15	-1.15
0.0526	-1.62	-1.62	0.1271	-1.14	-1.14
0.0537	-1.61	-1.61	0.1292	-1.13	-1.13
0.0548	-1.60	-1.60	0.1314	-1.12	-1.12
0.0559	-1.59	-1.59	0.1335	-1.11	-1.11
0.0571	-1.58	-1.58	0.1357	-1.10	-1.10
0.0582	-1.57	-1.57	0.1379	-1.09	-1.09
0.0594	-1.56	-1.56	0.1401	-1.08	-1.08
0.0606	-1.55	-1.55	0.1423	-1.07	-1.07
0.0618	-1.54	-1.54	0.1446	-1.06	-1.06
0.0630	-1.53	-1.53	0.1469	-1.05	-1.05
0.0643	-1.52	-1.52	0.1492	-1.04	-1.04
0.0655	-1.51	-1.51	0.1515	-1.03	-1.03
0.0668	-1.50	-1.50	0.1539	-1.02	-1.02
0.0681	-1.49	-1.49	0.1562	-1.01	-1.01
0.0694	-1.48	-1.48	0.1587	-1.00	-1.00
0.0708	-1.47	-1.47	0.1611	-0.99	-0.99
0.0721	-1.46	-1.46	0.1635	-0.98	-0.98
0.0735	-1.45	-1.45	0.1660	-0.97	-0.97
0.0749	-1.44	-1.44	0.1685	-0.96	-0.96
0.0764	-1.43	-1.43	0.1711	-0.95	-0.95

Area Under Normal curve	Value of Z From Program	Value of Z From Excel	Area Under Normal curve	Value of Z From Program	Value of Z From Excel
0.1736	-0.94	-0.94	0.3121	-0.49	-0.49
0.1762	-0.93	-0.93	0.3156	-0.48	-0.48
0.1788	-0.92	-0.92	0.3192	-0.47	-0.47
0.1814	-0.91	-0.91	0.3228	-0.46	-0.46
0.1841	-0.90	-0.90	0.3264	-0.45	-0.45
0.1867	-0.89	-0.89	0.3300	-0.44	-0.44
0.1894	-0.88	-0.88	0.3336	-0.43	-0.43
0.1922	-0.87	-0.87	0.3372	-0.42	-0.42
0.1949	-0.86	-0.86	0.3409	-0.41	-0.41
0.1977	-0.85	-0.85	0.3446	-0.40	-0.40
0.2005	-0.84	-0.84	0.3483	-0.39	-0.39
0.2033	-0.83	-0.83	0.3520	-0.38	-0.38
0.2061	-0.82	-0.82	0.3557	-0.37	-0.37
0.2090	-0.81	-0.81	0.3594	-0.36	-0.36
0.2119	-0.80	-0.80	0.3632	-0.35	-0.35
0.2148	-0.79	-0.79	0.3669	-0.34	-0.34
0.2177	-0.78	-0.78	0.3707	-0.33	-0.33
0.2206	-0.77	-0.77	0.3745	-0.32	-0.32
0.2236	-0.76	-0.76	0.3783	-0.31	-0.31
0.2266	-0.75	-0.75	0.3821	-0.30	-0.30
0.2296	-0.74	-0.74	0.3859	-0.29	-0.29
0.2327	-0.73	-0.73	0.3897	-0.28	-0.28
0.2358	-0.72	-0.72	0.3936	-0.27	-0.27
0.2389	-0.71	-0.71	0.3974	-0.26	-0.26
0.2420	-0.70	-0.70	0.4013	-0.25	-0.25
0.2451	-0.69	-0.69	0.4052	-0.24	-0.24
0.2483	-0.68	-0.68	0.4090	-0.23	-0.23
0.2514	-0.67	-0.67	0.4129	-0.22	-0.22
0.2546	-0.66	-0.66	0.4168	-0.21	-0.21
0.2578	-0.65	-0.65	0.4207	-0.20	-0.20
0.2611	-0.64	-0.64	0.4247	-0.19	-0.19
0.2643	-0.63	-0.63	0.4286	-0.18	-0.18
0.2676	-0.62	-0.62	0.4325	-0.17	-0.17
0.2709	-0.61	-0.61	0.4364	-0.16	-0.16
0.2743	-0.60	-0.60	0.4404	-0.15	-0.15
0.2776	-0.59	-0.59	0.4443	-0.14	-0.14
0.2810	-0.58	-0.58	0.4483	-0.13	-0.13
0.2843	-0.57	-0.57	0.4522	-0.12	-0.12
0.2877	-0.56	-0.56	0.4562	-0.11	-0.11
0.2912	-0.55	-0.55	0.4602	-0.10	-0.10
0.2946	-0.54	-0.54	0.4641	-0.09	-0.09
0.2981	-0.53	-0.53	0.4681	-0.08	-0.08
0.3015	-0.52	-0.52	0.4721	-0.07	-0.07
0.3050	-0.51	-0.51	0.4761	-0.06	-0.06
0.3085	-0.50	-0.50	0.4801	-0.05	-0.05

Area Under Normal curve	Value of Z From Program	Value of Z From Excel	Area Under Normal curve	Value of Z From Program	Value of Z From Excel
0.4840	-0.04	-0.04	0.6591	0.41	0.41
0.4880	-0.03	-0.03	0.6628	0.42	0.42
0.4920	-0.02	-0.02	0.6664	0.43	0.43
0.4960	-0.01	-0.01	0.6700	0.44	0.44
0.5000	0.00	0.00	0.6736	0.45	0.45
0.5040	0.01	0.01	0.6772	0.46	0.46
0.5080	0.02	0.02	0.6808	0.47	0.47
0.5120	0.03	0.03	0.6844	0.48	0.48
0.5160	0.04	0.04	0.6879	0.49	0.49
0.5199	0.05	0.05	0.6915	0.50	0.50
0.5239	0.06	0.06	0.6950	0.51	0.51
0.5279	0.07	0.07	0.6985	0.52	0.52
0.5319	0.08	0.08	0.7019	0.53	0.53
0.5359	0.09	0.09	0.7054	0.54	0.54
0.5398	0.10	0.10	0.7088	0.55	0.55
0.5438	0.11	0.11	0.7123	0.56	0.56
0.5478	0.12	0.12	0.7157	0.57	0.57
0.5517	0.13	0.13	0.7190	0.58	0.58
0.5557	0.14	0.14	0.7224	0.59	0.59
0.5596	0.15	0.15	0.7257	0.60	0.60
0.5636	0.16	0.16	0.7291	0.61	0.61
0.5675	0.17	0.17	0.7324	0.62	0.62
0.5714	0.18	0.18	0.7357	0.63	0.63
0.5753	0.19	0.19	0.7389	0.64	0.64
0.5793	0.20	0.20	0.7422	0.65	0.65
0.5832	0.21	0.21	0.7454	0.66	0.66
0.5871	0.22	0.22	0.7486	0.67	0.67
0.5910	0.23	0.23	0.7517	0.68	0.68
0.5948	0.24	0.24	0.7549	0.69	0.69
0.5987	0.25	0.25	0.7580	0.70	0.70
0.6026	0.26	0.26	0.7611	0.71	0.71
0.6064	0.27	0.27	0.7642	0.72	0.72
0.6103	0.28	0.28	0.7673	0.73	0.73
0.6141	0.29	0.29	0.7704	0.74	0.74
0.6179	0.30	0.30	0.7734	0.75	0.75
0.6217	0.31	0.31	0.7764	0.76	0.76
0.6255	0.32	0.32	0.7794	0.77	0.77
0.6293	0.33	0.33	0.7823	0.78	0.78
0.6331	0.34	0.34	0.7852	0.79	0.79
0.6368	0.35	0.35	0.7881	0.80	0.80
0.6406	0.36	0.36	0.7910	0.81	0.81
0.6443	0.37	0.37	0.7939	0.82	0.82
0.6480	0.38	0.38	0.7967	0.83	0.83
0.6517	0.39	0.39	0.7995	0.84	0.84
0.6554	0.40	0.40	0.8023	0.85	0.85

Area Under Normal curve	Value of Z From Program	Value of Z From Excel	Area Under Normal curve	Value of Z From Program	Value of Z From Excel
0.8051	0.86	0.86	0.9049	1.31	1.31
0.8078	0.87	0.87	0.9066	1.32	1.32
0.8106	0.88	0.88	0.9082	1.33	1.33
0.8133	0.89	0.89	0.9099	1.34	1.34
0.8159	0.90	0.90	0.9115	1.35	1.35
0.8186	0.91	0.91	0.9131	1.36	1.36
0.8212	0.92	0.92	0.9147	1.37	1.37
0.8238	0.93	0.93	0.9162	1.38	1.38
0.8264	0.94	0.94	0.9177	1.39	1.39
0.8289	0.95	0.95	0.9192	1.40	1.40
0.8315	0.96	0.96	0.9207	1.41	1.41
0.8340	0.97	0.97	0.9222	1.42	1.42
0.8365	0.98	0.98	0.9236	1.43	1.43
0.8389	0.99	0.99	0.9251	1.44	1.44
0.8413	1.00	1.00	0.9265	1.45	1.45
0.8438	1.01	1.01	0.9279	1.46	1.46
0.8461	1.02	1.02	0.9292	1.47	1.47
0.8485	1.03	1.03	0.9306	1.48	1.48
0.8508	1.04	1.04	0.9319	1.49	1.49
0.8531	1.05	1.05	0.9332	1.50	1.50
0.8554	1.06	1.06	0.9345	1.51	1.51
0.8577	1.07	1.07	0.9357	1.52	1.52
0.8599	1.08	1.08	0.9370	1.53	1.53
0.8621	1.09	1.09	0.9382	1.54	1.54
0.8643	1.10	1.10	0.9394	1.55	1.55
0.8665	1.11	1.11	0.9406	1.56	1.56
0.8686	1.12	1.12	0.9418	1.57	1.57
0.8708	1.13	1.13	0.9429	1.58	1.58
0.8729	1.14	1.14	0.9441	1.59	1.59
0.8749	1.15	1.15	0.9452	1.60	1.60
0.8770	1.16	1.16	0.9463	1.61	1.61
0.8790	1.17	1.17	0.9474	1.62	1.62
0.8810	1.18	1.18	0.9484	1.63	1.63
0.8830	1.19	1.19	0.9495	1.64	1.64
0.8849	1.20	1.20	0.9505	1.65	1.65
0.8869	1.21	1.21	0.9515	1.66	1.66
0.8888	1.22	1.22	0.9525	1.67	1.67
0.8907	1.23	1.23	0.9535	1.68	1.68
0.8925	1.24	1.24	0.9545	1.69	1.69
0.8944	1.25	1.25	0.9554	1.70	1.70
0.8962	1.26	1.26	0.9564	1.71	1.71
0.8980	1.27	1.27	0.9573	1.72	1.72
0.8997	1.28	1.28	0.9582	1.73	1.73
0.9015	1.29	1.29	0.9591	1.74	1.74
0.9032	1.30	1.30	0.9599	1.75	1.75

ภาคผนวก B
การคำนวณคะแนนทีปกติ และตัดเกรดโดยใช้ โปรแกรมสำเร็จรูป Excel

ในภาคผนวก B นี้เป็นการนำโปรแกรมสำเร็จรูป Excel มาคำนวณคะแนนทีปกติ และตัดเกรด เพื่อเปรียบเทียบกับผลลัพธ์ที่ได้จากโปรแกรม TScoreCal

ข้อมูลที่ใช้ในครั้งนี้เป็นคะแนนดิบของผลการสอบวิชาฟิสิกส์ 1 ปีการศึกษา 1/2550 ของสาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคล ธัญบุรี จำนวน 1,413 คน ในที่นี้จะใช้ Excel รุ่น 2003 ในการคำนวณ

ลำดับขั้นตอนการคำนวณคะแนนทีปกติ โดยใช้ Excel มีดังนี้

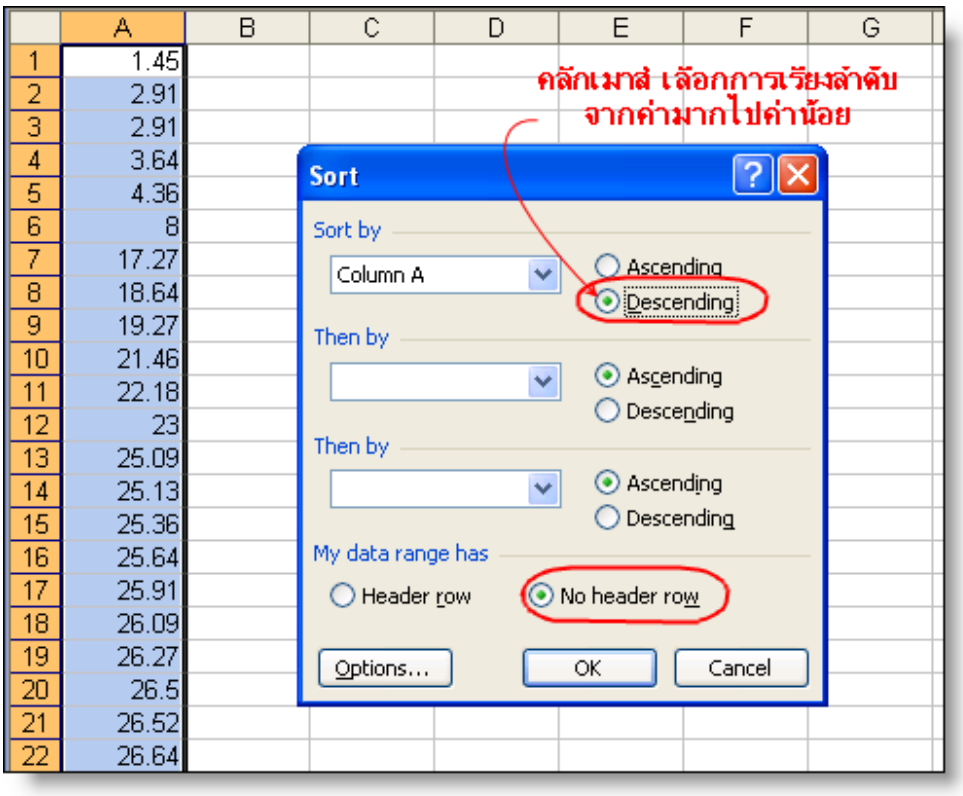
1. เมื่อเรียกใช้โปรแกรม Excel แล้ว ให้นำคะแนนดิบ เข้าเก็บไว้ในคอลัมน์ A ดังรูปที่ B1 (คะแนนดิบนี้เก็บไว้ในไฟล์ที่ชื่อ p1_1_50.txt สามารถใช้ Nodepad เปิด แล้ว copy และ paste มายัง Excel)

	A	B	C	D	E	F
10	21.46					
11	22.18					
12	23					
13	25.09					
14	25.13					
15	25.36					
16	25.64					
17	25.91					
18	26.09					
19	26.27					
20	26.5					
21	26.52					
22	26.64					
23	26.86					
24	26.91					
25	26.91					
26	27.18					
27	27.27					
28	27.27					
29	27.27					
30	27.41					
31	27.59					
32	27.82					
33	27.84					
34	27.86					
35	28					

รูปที่ B1 นำคะแนนดิบเก็บไว้ในคอลัมน์ A

2. เรียงลำดับข้อมูลในคอลัมน์ A ในที่นี้จะเรียงลำดับจากคะแนนสูงไปคะแนนต่ำ

เริ่มต้นด้วยการใช้เมาส์ลากระบายคอลัมน์ A ตั้งแต่แถวที่ 1 จนถึงแถวที่ 1413 จากนั้นคลิกที่เมนู Data -> Sort… จะปรากฏหน้าต่างขึ้นมาดังรูปที่ B2



รูปที่ B2 แสดงหน้าต่าง Sort ข้อมูล

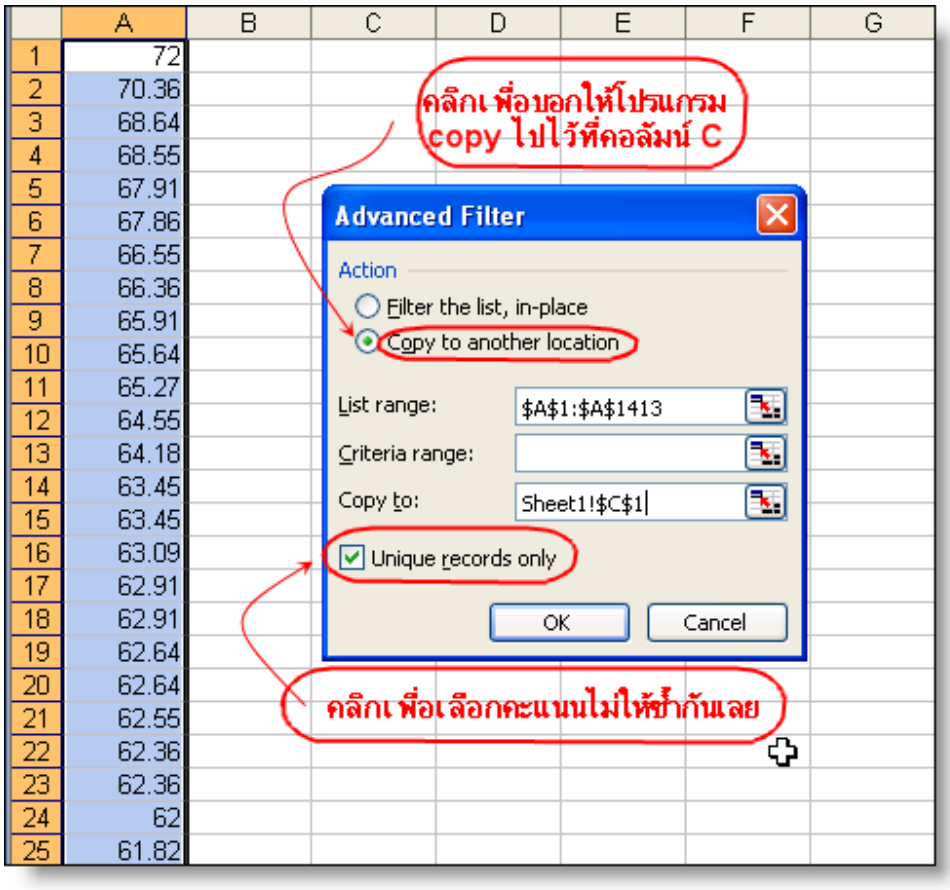
ที่หัวข้อ Sort by ในหน้าต่าง Sort ให้คลิกที่ปุ่ม Descending

ที่หัวข้อ My data range has ให้คลิกที่ No header row

คลิกปุ่ม OK โปรแกรมจะทำการเรียงลำดับคะแนนจากค่าสูงสุดไปยังค่าต่ำสุด

3. ข้อมูลในคอลัมน์ A ยังคงถูกระบายทึบ (ถ้ายังไม่ถูกระบาย ให้ลากเมาส์ระบายข้อมูลในคอลัมน์ Aทั้งหมดอีกครั้ง)

คลิกที่เมนู Data -> Filter -> advance Filter เพื่อทำการคัดเลือกคะแนนที่ไม่ซ้ำกันเลย และเก็บไว้ในคอลัมน์ C



รูปที่ B3 เลือกเฉพาะคะแนนที่ค่าไม่ซ้ำกันเลย เก็บไว้ในคอลัมน์ C

คลิกปุ่ม OK ที่คอลัมน์ C จะปรากฏคะแนนจำนวน 415 แถว ซึ่งเป็นคะแนนที่มีค่าไม่ซ้ำกัน ดังรูปที่ B4

	A	B	C	D	E	F
388	42.55		27.82			
389	42.55		27.59			
390	42.55		27.41			
391	42.36		27.27			
392	42.36		27.18			
393	42.36		26.91			
394	42.36		26.86			
395	42.36		26.64			
396	42.36		26.52			
397	42.27		26.5			
398	42.27		26.27			
399	42.23		26.09			
400	42.18		25.91			
401	42.18		25.64			
402	42.18		25.36			
403	42.18		25.13			
404	42.18		25.09			
405	42.18		23			
406	42.18		22.18			
407	42.09		21.46			
408	42.05		19.27			
409	42		18.64			
410	42		17.27			
411	42		8			
412	42		4.36			
413	42		3.64			
414	41.91		2.91			
415	41.91		1.45			
416	41.85					
417	41.82					
418	41.82					

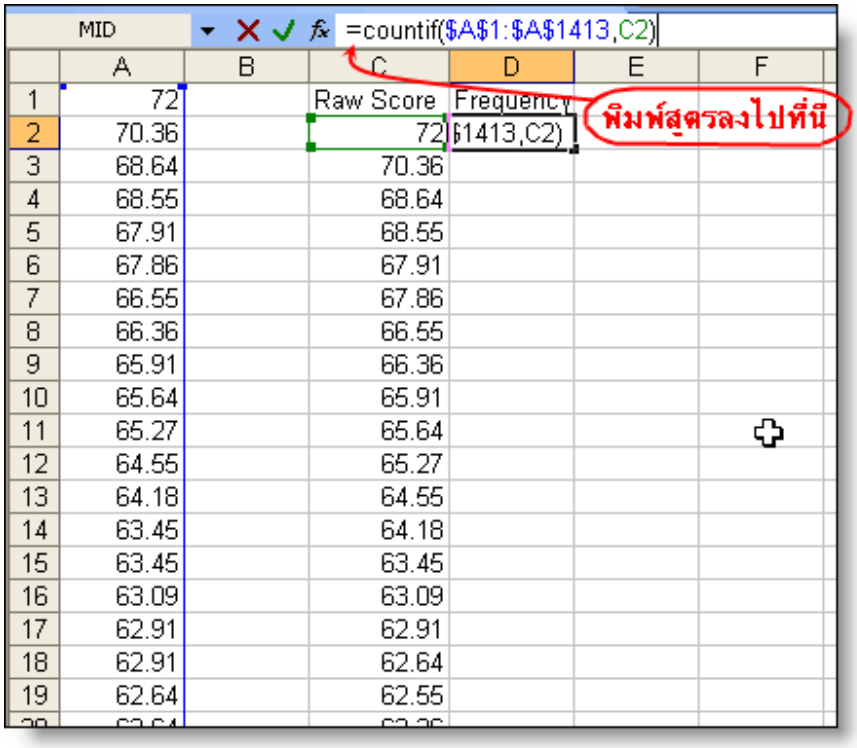
รูปที่ B4 คะแนนในคอลัมน์ C จะมีค่าไม่ซ้ำกัน

4. หาความถี่ของคะแนนดิบแต่ละค่า ที่คอลัมน์ C เราอาจเว้นที่ว่างสำหรับเขียนหัวคอลัมน์ซึ่งต่อไปจะมีเพิ่มอีกหลายคอลัมน์ เพื่อจะได้ไม่สับสนในภายหลัง

ที่คอลัมน์ D ให้พิมพ์สูตรต่อไปนี้ลงในเซลล์ D2

=countif($A$1:$A$1413,C2)

ดังรูปที่ B5



	A	B	C	D	E	F
1	72		Raw Score	Frequency		
2	70.36		72	$1413,C2)		
3	68.64		70.36			
4	68.55		68.64			
5	67.91		68.55			
6	67.86		67.91			
7	66.55		67.86			
8	66.36		66.55			
9	65.91		66.36			
10	65.64		65.91			
11	65.27		65.64			
12	64.55		65.27			
13	64.18		64.55			
14	63.45		64.18			
15	63.45		63.45			
16	63.09		63.09			
17	62.91		62.91			
18	62.91		62.64			
19	62.64		62.55			

รูปที่ B5 ใส่สูตรลงไปในเซลล์ C2

จากนั้นให้ copy สูตรในเซลล์ C2 ลงมาใส่ในทุกแถวของคะแนนทุกค่า เครื่องหมาย $ หน้าชื่อคอลัมน์และหน้าตัวเลขบอกแถว เป็นการบอกให้ Excel รู้ว่าช่วงที่กำหนดไว้ คือ A1 ถึง A1413 นี้จะคงที่เสมอ ไม่ว่าสูตรจะถูก copy ไปยังเซลล์ใด

ความหมายของสูตร countif($A$1:$A$1413,C2) คือ ให้นับจำนวนคะแนนที่เท่ากับคะแนนในเซลล์ C2 (ในที่นี้คือ 72) ในช่วงตั้งแต่ A1 ถึง A1413 มีอยู่กี่ตัว เป็นการหาความถี่ของคะแนนแต่ละค่านั่นเอง

D330 =COUNTIF($A$1:$A$1413,C330)

	A	B	C	D	E	F
1	72		Raw Score	Frequency		
2	70.36		72	1		
3	68.64		70.36	1		
4	68.55		68.64	1		
5	67.91		68.55	1		
6	67.86		67.91	1		
7	66.55		67.86	1		
8	66.36		66.55	1		
9	65.91		66.36	1		
10	65.64		65.91	1		
11	65.27		65.64	1		
12	64.55		65.27	1		
13	64.18		64.55	1		
14	63.45		64.18	1		
15	63.45		63.45	2		
16	63.09		63.09	1		
17	62.91		62.91	2		
18	62.91		62.64	2		
19	62.64		62.55	1		
20	62.64		62.36	2		
21	62.55		62	1		
22	62.36		61.82	1		
23	62.36		61.64	1		
24	62		61.18	1		
25	61.82		60.36	1		

รูปที่ B6 แสดงค่าความถี่ที่ได้

5. หาความถี่สะสม(Cumulative frequency, cf) หาได้จากนำความถี่สะสมของคะแนนที่อยู่ต่ำกว่า 1 แถว บวกด้วยความถี่ของคะแนนในแถวนั้น

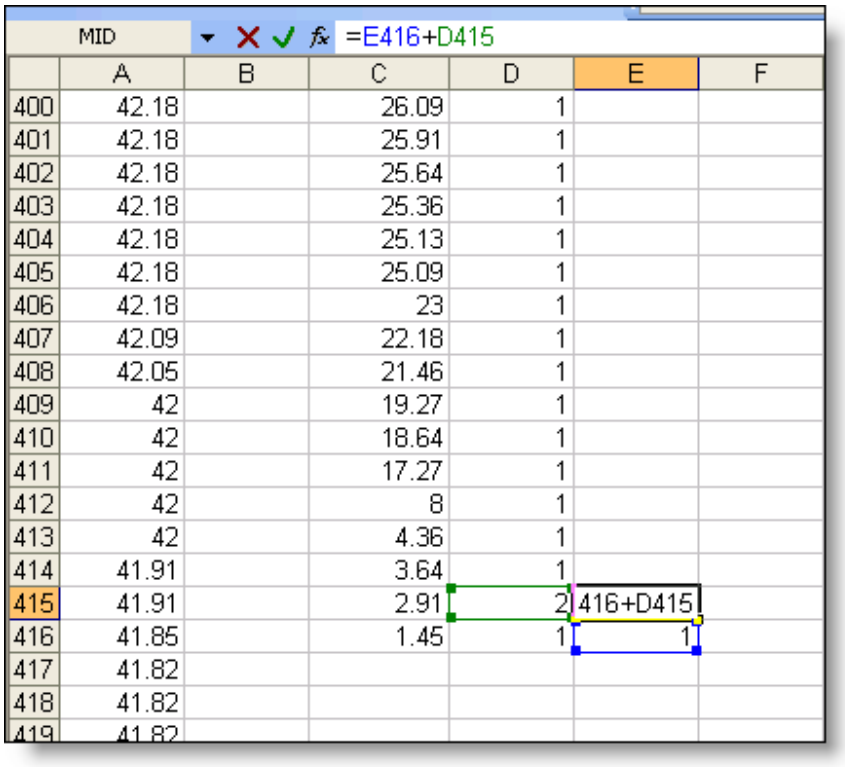
เลื่อน scrollbar ไปจนถึงคะแนนต่ำสุด ในที่นี้คือ แถวที่ 416 คะแนนดิบ 1.45

ที่เซลล์ E416 พิมพ์ 1 ลงไปเซลล์นี้ (1 นี้คือความถี่ของคะแนนดิบ 1.45)

ที่เซลล์ E415 พิมพ์สูตร = E416 + D415

Copy สูตรในเซลล์ E415 ลงไปในเซลล์ E414 จนถึง E2

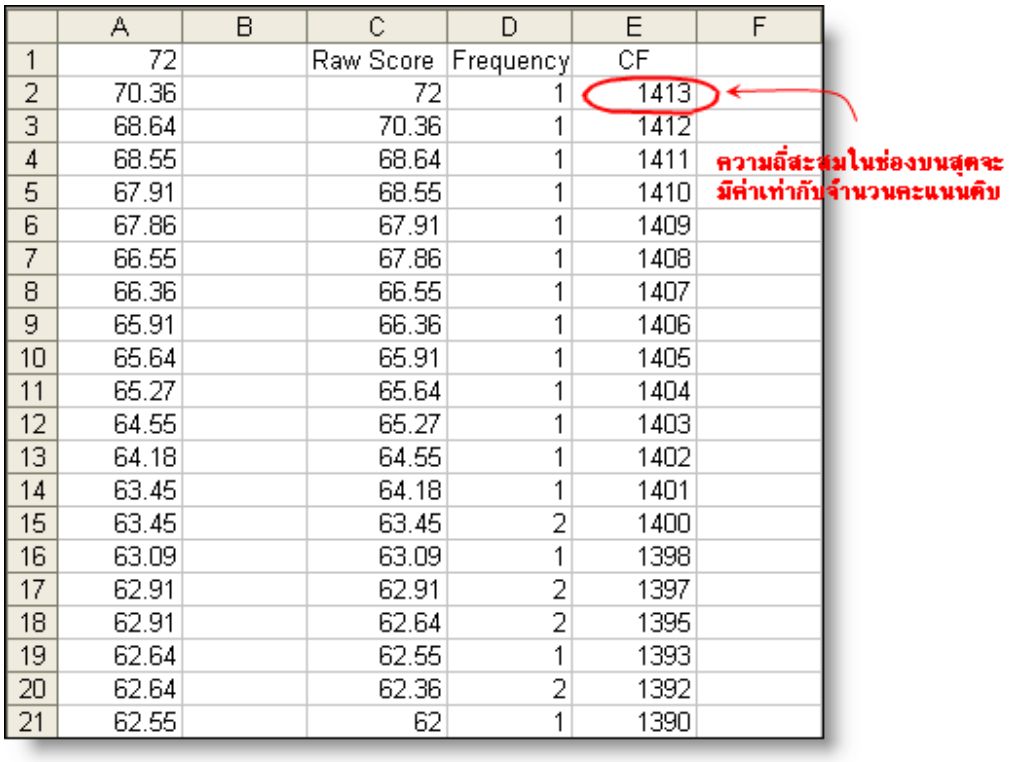
จะได้ผลลัพธ์ซึ่งเป็นค่าความถี่สะสมในคอลัมน์ E



รูปที่ B7 แสดงการเขียนสูตรหาความถี่สะสม

ถ้าทำได้ถูกต้อง จะสังเกตเห็นว่า เซลล์ E2 ซึ่งเป็นเซลล์แรกสุดของความถี่สะสม จะมีค่าเท่ากับจำนวนของคะแนนดิบ ในที่นี้คือ 1413 ดังรูปที่ B8

	A	B	C	D	E	F
1	72		Raw Score	Frequency	CF	
2	70.36		72	1	1413	
3	68.64		70.36	1	1412	
4	68.55		68.64	1	1411	
5	67.91		68.55	1	1410	
6	67.86		67.91	1	1409	
7	66.55		67.86	1	1408	
8	66.36		66.55	1	1407	
9	65.91		66.36	1	1406	
10	65.64		65.91	1	1405	
11	65.27		65.64	1	1404	
12	64.55		65.27	1	1403	
13	64.18		64.55	1	1402	
14	63.45		64.18	1	1401	
15	63.45		63.45	2	1400	
16	63.09		63.09	1	1398	
17	62.91		62.91	2	1397	
18	62.91		62.64	2	1395	
19	62.64		62.55	1	1393	
20	62.64		62.36	2	1392	
21	62.55		62	1	1390	



รูปที่ B8 แสดงค่าความถี่สะสม (CF) ที่คำนวณได้

6. คำนวณหาเปอร์เซ็นต์ไทล์ของคะแนนดิบแต่ละคะแนน หาได้จากสูตร

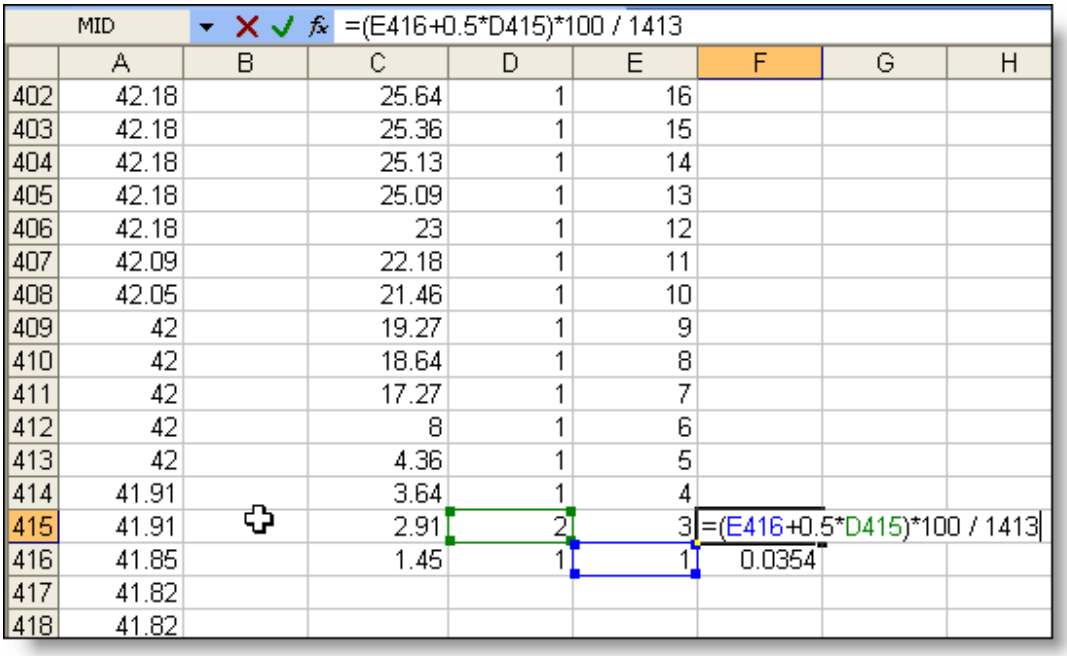
$$\text{Percentile} = \frac{(\text{CF ของคะแนนที่อยู่ต่ำกว่า 1 แถว} + 0.5\ \text{ความถี่ของคะแนนนั้น}) \times 100}{\text{จำนวนผู้เข้าสอบ}}$$

เลื่อน scrollbar ลงไปด้านล่างของตารางคำนวณ ใช้เมาส์คลิกที่เซลล์ F416

พิมพ์ 0.0354 ลงไปในเซลล์ F416 (ตัวเลขนี้ได้มาจาก (0.5)(1)*100 / (1413))

ที่เซลล์ F415 เขียนสูตรต่อไปนี้ลงในเซลล์

= (E416+0.5*D415)*100 / 1413



	A	B	C	D	E	F	G	H
402	42.18		25.64	1	16			
403	42.18		25.36	1	15			
404	42.18		25.13	1	14			
405	42.18		25.09	1	13			
406	42.18		23	1	12			
407	42.09		22.18	1	11			
408	42.05		21.46	1	10			
409	42		19.27	1	9			
410	42		18.64	1	8			
411	42		17.27	1	7			
412	42		8	1	6			
413	42		4.36	1	5			
414	41.91		3.64	1	4			
415	41.91		2.91	2	3	=(E416+0.5*D415)*100 / 1413		
416	41.85		1.45	1	1	0.0354		
417	41.82							
418	41.82							

รูปที่ B9 เขียนสูตรเพื่อหา Percentile

Copy สูตรจากเซลล์ F415 ไปไว้ในเซลล์ F414 ขึ้นไปเรื่อย ๆ จนถึง เซลล์ F2 จะได้ค่าเปอร์เซ็นต์ไทล์ เก็บไว้ในคอลัมน์ F

ที่คอลัมน์ ปรับตำแหน่งทศนิยมของ Percentile ให้เหลือ 2 ตำแหน่ง

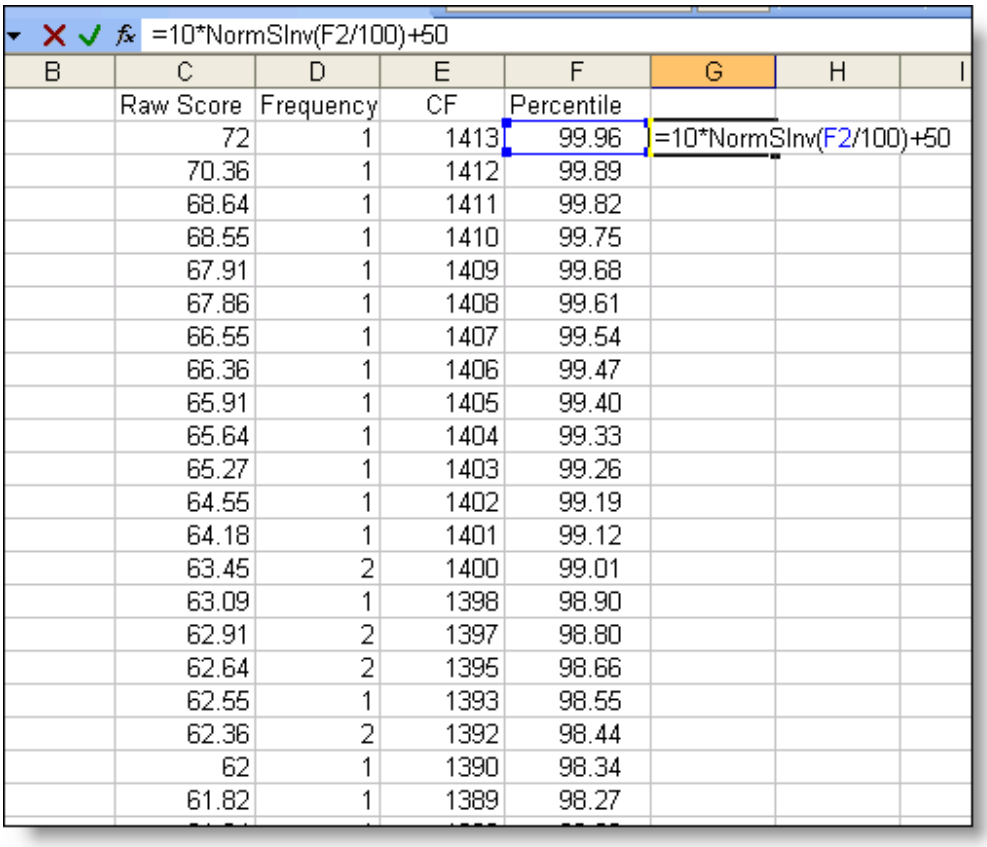
	A	B	C	D	E	F
1	72		Raw Score	Frequency	CF	Percentile
2	70.36		72	1	1413	99.96
3	68.64		70.36	1	1412	99.89
4	68.55		68.64	1	1411	99.82
5	67.91		68.55	1	1410	99.75
6	67.86		67.91	1	1409	99.68
7	66.55		67.86	1	1408	99.61
8	66.36		66.55	1	1407	99.54
9	65.91		66.36	1	1406	99.47
10	65.64		65.91	1	1405	99.40
11	65.27		65.64	1	1404	99.33
12	64.55		65.27	1	1403	99.26
13	64.18		64.55	1	1402	99.19
14	63.45		64.18	1	1401	99.12
15	63.45		63.45	2	1400	99.01
16	63.09		63.09	1	1398	98.90
17	62.91		62.91	2	1397	98.80
18	62.91		62.64	2	1395	98.66
19	62.64		62.55	1	1393	98.55
20	62.64		62.36	2	1392	98.44
21	62.55		62	1	1390	98.34
22	62.36		61.82	1	1389	98.27
23	62.36		61.64	1	1388	98.20
24	62		61.18	1	1387	98.12
25	61.82		60.36	1	1386	98.05

รูปที่ B10 แสดงเปอร์เซ็นต์ไทล์ในคอลัมน์ F หลังจากปรับเลขทศนิยมให้เหลือ 2 ตำแหน่ง

7. คำนวณหาคะแนนทีปกติ โดยคำนวณจากพื้นที่ใต้เส้นโค้งปกติ เรียกใช้ฟังก์ชัน NormSInv( ) ซึ่งเป็นฟังก์ชัน Built in ในโปรแกรม Excel

คลิกเมาส์ที่เซลล์ G2 พิมพ์สูตรต่อไปนี้ลงไปในเซลล์นี้

= 10*NormSInv(F2/100) +50



รูปที่ B11 แสดงการใส่สูตรเพื่อหาคะแนนทีปกติ

Copy สูตรจากเซลล์ G2 ไปใส่ในเซลล์ G3 จนถึงเซลล์ G416

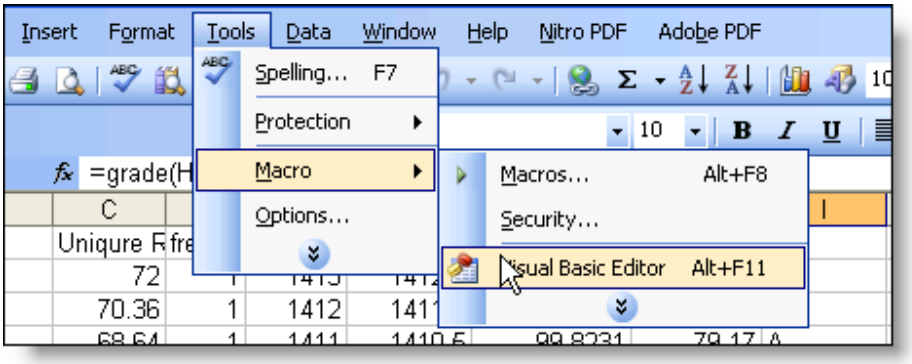
จัดรูปแบบตัวเลขในคอลัมน์ G2 ให้เหลือทศนิยมเพียง 2 ตำแหน่ง

B	C	D	E	F	G
	Raw Score	Frequency	CF	Percentile	Normalized T
	72	1	1413	99.96	83.87
	70.36	1	1412	99.89	80.72
	68.64	1	1411	99.82	79.17
	68.55	1	1410	99.75	78.10
	67.91	1	1409	99.68	77.28
	67.86	1	1408	99.61	76.61
	66.55	1	1407	99.54	76.05
	66.36	1	1406	99.47	75.55
	65.91	1	1405	99.40	75.11
	65.64	1	1404	99.33	74.72
	65.27	1	1403	99.26	74.36
	64.55	1	1402	99.19	74.03
	64.18	1	1401	99.12	73.72
	63.45	2	1400	99.01	73.30
	63.09	1	1398	98.90	72.91
	62.91	2	1397	98.80	72.56
	62.64	2	1395	98.66	72.13
	62.55	1	1393	98.55	71.83
	62.36	2	1392	98.44	71.55
	62	1	1390	98.34	71.29
	61.82	1	1389	98.27	71.12
	61.64	1	1388	98.20	70.96
	61.18	1	1387	98.12	70.80
	60.36	1	1386	98.05	70.65

รูปที่ B12 แสดงคะแนน T ปกติที่คำนวณได้ ในคอลัมน์ G

8. ขั้นต่อไปคือการตัดเกรด ในที่นี้จะตัดเกรดเพียง 5 เกรด คือ A, B, C, D และ F จะให้โปรแกรม Excel ใส่เกรดให้ โดยเขียนคำสั่งใน Visual Basic Editor ดังต่อไปนี้

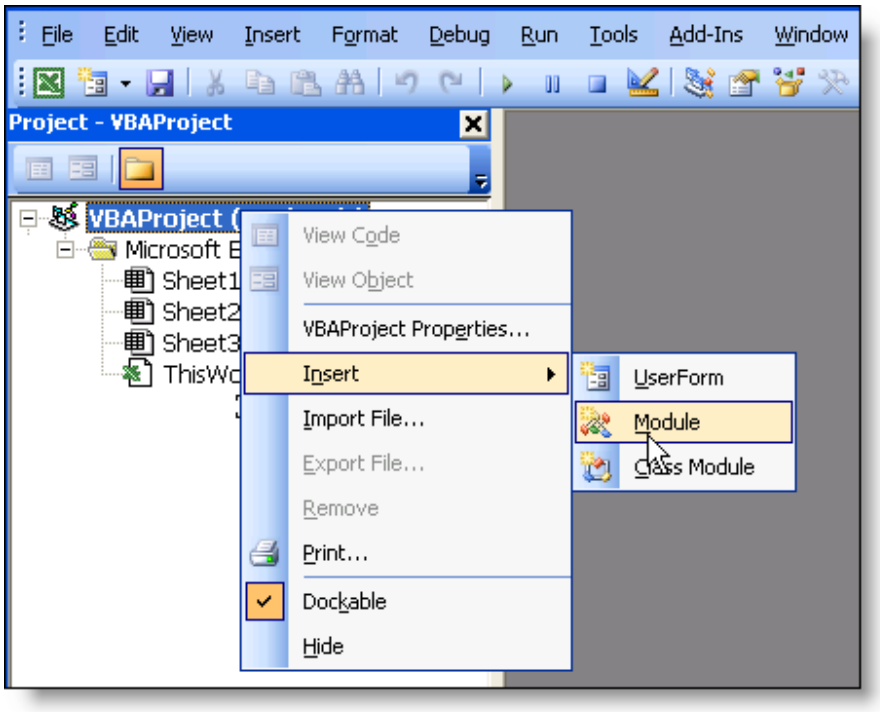
คลิกเมนูที่คำสั่ง Tools -> Marco -> Visual Basic Editor หรือใช้คีย์ลัด คือ ALT + F11



รูปที่ B13 แสดงการเข้า Visual Basic Editor

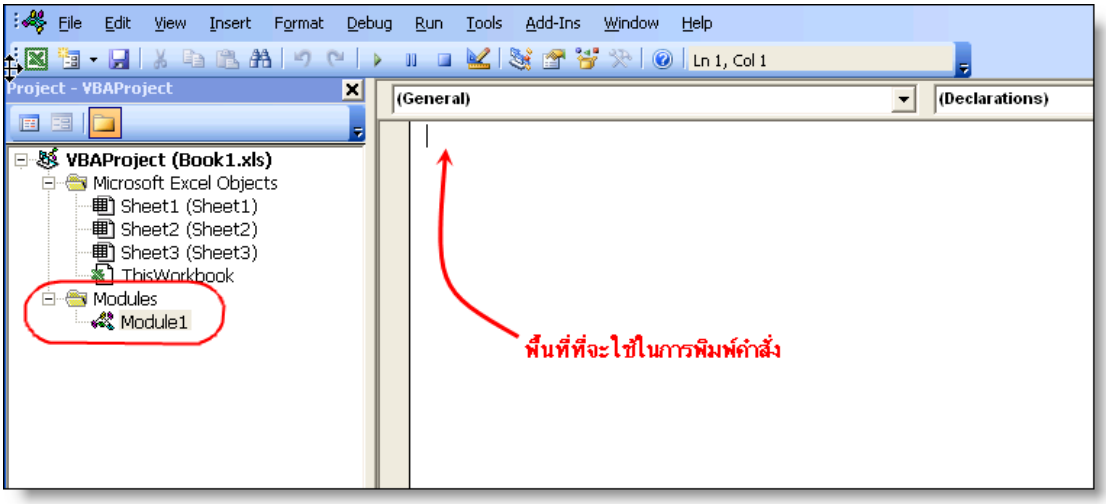
หน้าจอภาพจะเปลี่ยนไปเป็นหน้าต่างของ Visual Basic Editor

คลิกเมาส์ปุ่มขวา ที่ชื่อโปรเจกต์ (ในที่นี้คือ VBAProject (Book1.xls)) แล้วคลิกที่ Insert -> Module ดังรูปที่ B14



รูปที่ B14 สร้าง Module สำหรับ Book1.xls

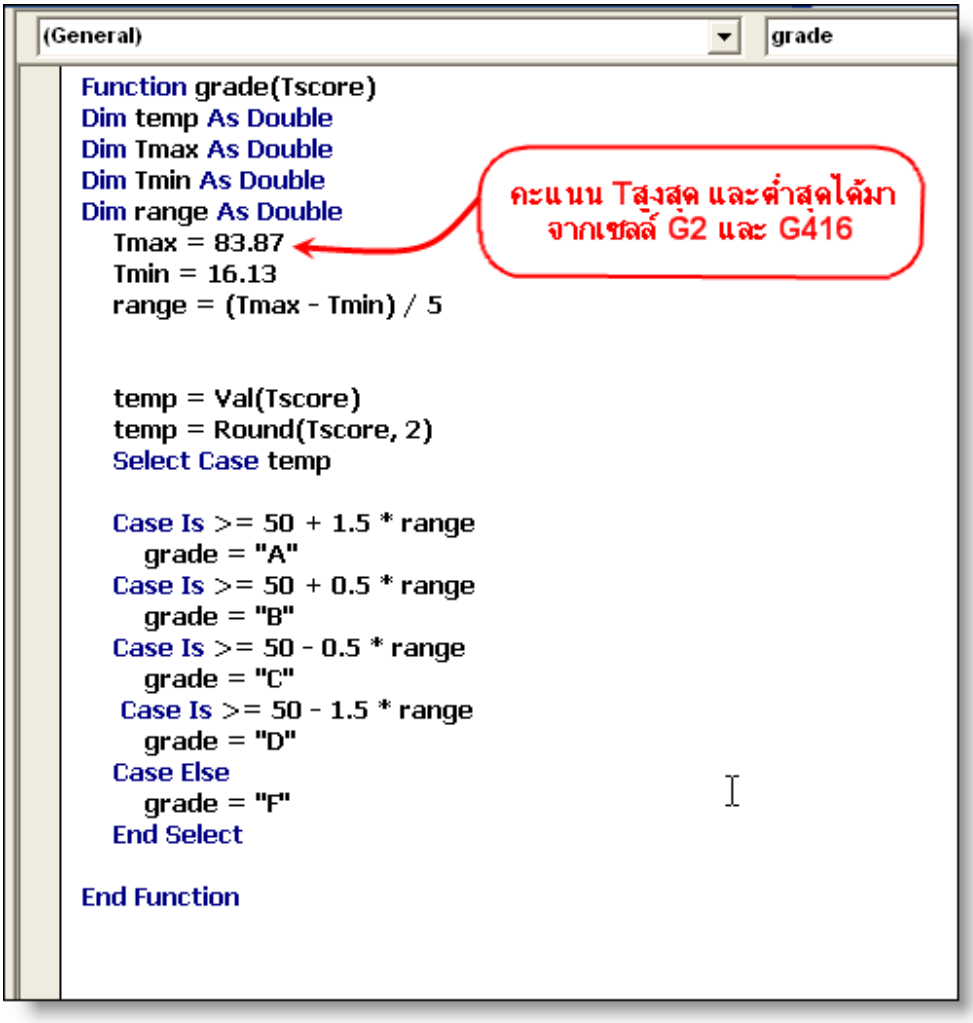
จะได้ Moduleใหม่ชื่อว่า Module1 พร้อมกับมีหน้าต่างว่าง ขึ้นมาสำหรับพิมพ์คำสั่ง



รูปที่ 15 แสดง Visual Basic Editor สำหรับพิมพ์คำสั่ง

พิมพ์ฟังก์ชันชื่อว่า Grade ( ) เพื่อให้โปรแกรมสามารถกำหนดเกรดให้อย่างอัตโนมัติ พิมพ์คำสั่งลงไป ตามรูปที่ B16 ฟังก์ชัน Grade( ) นี้ใช้ได้เฉพาะ worksheet นี้เท่านั้น เพราะค่า Tmax และ Tmin ได้มาจากข้อมูลในตารางนี้ ถ้าต้องการใช้ฟังก์ชัน Grade ( ) กับ worksheet อื่น ๆ ให้เปลี่ยนค่า Tmax และ Tmin ให้สอดคล้องกับค่าที่ได้ใน worksheet นั้น ๆ

```
Function grade(Tscore)
Dim temp As Double
Dim Tmax As Double
Dim Tmin As Double
Dim range As Double
    Tmax = 83.87
    Tmin = 16.13
    range = (Tmax - Tmin) / 5


    temp = Val(Tscore)
    temp = Round(Tscore, 2)
    Select Case temp

    Case Is >= 50 + 1.5 * range
        grade = "A"
    Case Is >= 50 + 0.5 * range
        grade = "B"
    Case Is >= 50 - 0.5 * range
        grade = "C"
     Case Is >= 50 - 1.5 * range
        grade = "D"
    Case Else
        grade = "F"
    End Select

End Function
```

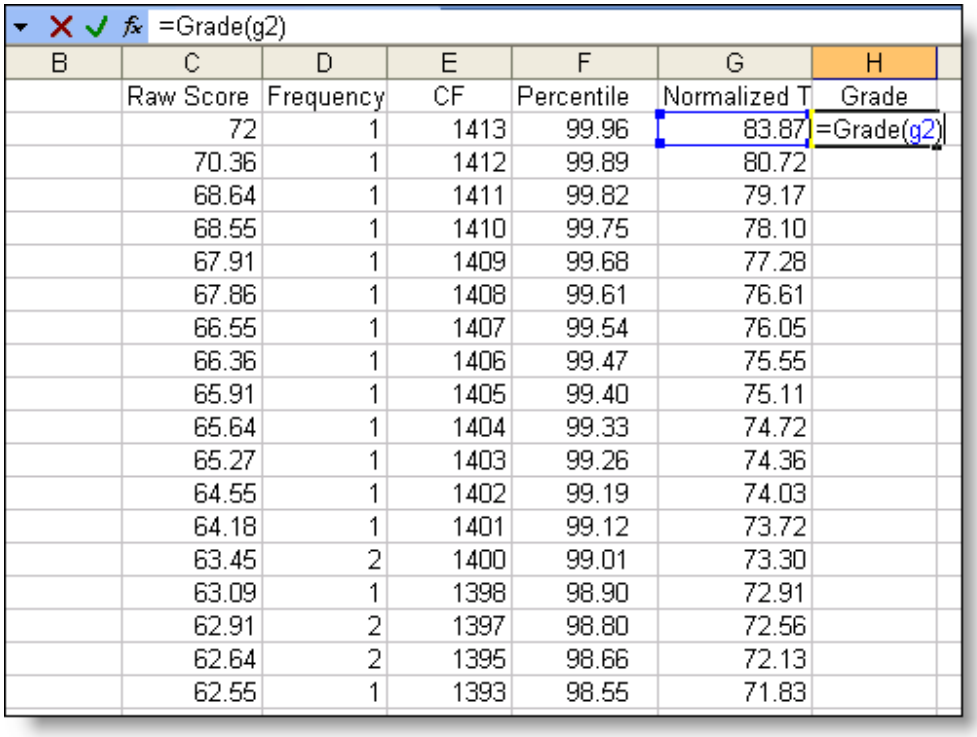


รูปที่ B16 คำสั่งที่พิมพ์ลงใน editor ของ VBA (Visual Basic Application)

Save ฟังก์ชัน Grade ( ) แล้ว ให้กลับมาที่หน้าต่าง worksheet เดิมที่ได้คำนวณคะแนนที่ปกติ

ที่ เซลล์ H2 พิมพ์สูตรลงไปในเซลล์ดังนี้

= Grade(G2)



B	C	D	E	F	G	H
	Raw Score	Frequency	CF	Percentile	Normalized T	Grade
	72	1	1413	99.96	83.87	=Grade(g2)
	70.36	1	1412	99.89	80.72	
	68.64	1	1411	99.82	79.17	
	68.55	1	1410	99.75	78.10	
	67.91	1	1409	99.68	77.28	
	67.86	1	1408	99.61	76.61	
	66.55	1	1407	99.54	76.05	
	66.36	1	1406	99.47	75.55	
	65.91	1	1405	99.40	75.11	
	65.64	1	1404	99.33	74.72	
	65.27	1	1403	99.26	74.36	
	64.55	1	1402	99.19	74.03	
	64.18	1	1401	99.12	73.72	
	63.45	2	1400	99.01	73.30	
	63.09	1	1398	98.90	72.91	
	62.91	2	1397	98.80	72.56	
	62.64	2	1395	98.66	72.13	
	62.55	1	1393	98.55	71.83	

รูปที่ B17 แสดงการเรียกใช้งานฟังก์ชัน Grade ( )

Copy สูตรจากเซลล์ H2 ไปยังเซลล์ H3 จนถึงเซลล์ H416 จะเห็นเกรดปรากฏในคอลัมน์ H

C	D	E	F	G	H
Raw Score	Frequency	CF	Percentile	Normalized T	Grade
72	1	1413	99.96	83.87	A
70.36	1	1412	99.89	80.72	A
68.64	1	1411	99.82	79.17	A
68.55	1	1410	99.75	78.10	A
67.91	1	1409	99.68	77.28	A
67.86	1	1408	99.61	76.61	A
66.55	1	1407	99.54	76.05	A
66.36	1	1406	99.47	75.55	A
65.91	1	1405	99.40	75.11	A
65.64	1	1404	99.33	74.72	A
65.27	1	1403	99.26	74.36	A
64.55	1	1402	99.19	74.03	A
64.18	1	1401	99.12	73.72	A
63.45	2	1400	99.01	73.30	A
63.09	1	1398	98.90	72.91	A
62.91	2	1397	98.80	72.56	A
62.64	2	1395	98.66	72.13	A
62.55	1	1393	98.55	71.83	A
62.36	2	1392	98.44	71.55	A
62	1	1390	98.34	71.29	A
61.82	1	1389	98.27	71.12	A
61.64	1	1388	98.20	70.96	A
61.18	1	1387	98.12	70.80	A
60.36	1	1386	98.05	70.65	A
60.18	1	1385	97.98	70.50	A
60	1	1384	97.91	70.36	A
59.27	1	1383	97.84	70.22	B
59.09	1	1382	97.77	70.09	B
58.91	2	1381	97.66	69.89	B
58.73	1	1379	97.56	69.70	B
58.36	1	1378	97.49	69.58	B
58.18	2	1377	97.38	69.40	B

รูปที่ B18 เกรดจะแสดงให้เห็นในคอลัมน์ H

9. เปรียบเทียบผลลัพธ์ที่ได้จากโปรแกรม Excel พบว่า ความถี่ ความถี่สะสม Percentile คะแนน T ปกติ ได้ค่าตรงกับค่าที่ได้จากคำนวณของโปรแกรม TScoreCal ทุกคอลัมน์ (ที่ทศนิยม 2 ตำแหน่ง) ขั้นตอนวิธีการตัดเกรดเมื่อเขียนคำสั่งให้ Excel ประมวลผล จะได้ผลลัพธ์ตรงกับที่คำนวณได้จากโปรแกรม TScoreCal เช่นกัน ดังรูปที่ 19 ด้านซ้ายมือ บริเวณพื้นที่สีขาวคือผลลัพธ์ที่ได้จากโปรแกรม Excel ด้านขวามือ บริเวณพื้นที่ที่แรเงาเป็นผลลัพธ์ที่ได้จากการคำนวณของโปรแกรม TScoreCal โดยใช้วิธี copy และ paste เข้ามาในโปรแกรม Excel ถ้าต้องการดูตารางที่ครบสมบูรณ์สามารถเปิดดูได้จากไฟล์ชื่อ p1_1_50.xls ที่บันทึกอยู่ในแผ่น CD ที่แนบมาพร้อมกับรายงานฉบับนี้

C	D	E	F	G	H	I	J	K	L	M	N
Raw Score	frequency	cf	Percentile	T score	Grade	Raw Score	frequency	cf	Percentile	T score	Grade
72	1	1413	99.96	83.87	A	72	1	1413	99.96	83.87	A
70.36	1	1412	99.89	80.72	A	70.36	1	1412	99.89	80.72	A
68.64	1	1411	99.82	79.17	A	68.64	1	1411	99.82	79.17	A
68.55	1	1410	99.75	78.10	A	68.55	1	1410	99.75	78.1	A
67.91	1	1409	99.68	77.28	A	67.91	1	1409	99.68	77.28	A
67.86	1	1408	99.61	76.61	A	67.86	1	1408	99.61	76.61	A
66.55	1	1407	99.54	76.05	A	66.55	1	1407	99.54	76.05	A
66.36	1	1406	99.47	75.55	A	66.36	1	1406	99.47	75.55	A
65.91	1	1405	99.40	75.11	A	65.91	1	1405	99.4	75.11	A
65.64	1	1404	99.33	74.72	A	65.64	1	1404	99.33	74.72	A
65.27	1	1403	99.26	74.36	A	65.27	1	1403	99.26	74.36	A
64.55	1	1402	99.19	74.03	A	64.55	1	1402	99.19	74.03	A
64.18	1	1401	99.12	73.72	A	64.18	1	1401	99.12	73.72	A
63.45	2	1400	99.01	73.30	A	63.45	2	1400	99.01	73.3	A
63.09	1	1398	98.90	72.91	A	63.09	1	1398	98.9	72.91	A
62.91	2	1397	98.80	72.56	A	62.91	2	1397	98.8	72.56	A
62.64	2	1395	98.66	72.13	A	62.64	2	1395	98.66	72.13	A
62.55	1	1393	98.55	71.83	A	62.55	1	1393	98.55	71.83	A
62.36	2	1392	98.44	71.55	A	62.36	2	1392	98.44	71.55	A
62	1	1390	98.34	71.29	A	62	1	1390	98.34	71.29	A
61.82	1	1389	98.27	71.12	A	61.82	1	1389	98.27	71.12	A
61.64	1	1388	98.20	70.96	A	61.64	1	1388	98.2	70.96	A
61.18	1	1387	98.12	70.80	A	61.18	1	1387	98.12	70.8	A
60.36	1	1386	98.05	70.65	A	60.36	1	1386	98.05	70.65	A
60.18	1	1385	97.98	70.50	A	60.18	1	1385	97.98	70.5	A
60	1	1384	97.91	70.36	A	60	1	1384	97.91	70.36	A
59.27	1	1383	97.84	70.22	B	59.27	1	1383	97.84	70.22	B
59.09	1	1382	97.77	70.09	B	59.09	1	1382	97.77	70.09	B
58.91	2	1381	97.66	69.89	B	58.91	2	1381	97.66	69.89	B
58.73	1	1379	97.56	69.70	B	58.73	1	1379	97.56	69.7	B
[illegible]											
[illegible]					D	[illegible]					D
27.84	1	33	2.30	30.05	D	27.84	1	33	2.3	30.05	D
27.82	1	32	2.23	29.91	D	27.82	1	32	2.23	29.91	D
27.59	1	31	2.16	29.78	D	27.59	1	31	2.16	29.78	D
27.41	1	30	2.09	29.64	F	27.41	1	30	2.09	29.64	F
27.27	3	29	1.95	29.35	F	27.27	3	29	1.95	29.35	F
27.18	1	26	1.80	29.04	F	27.18	1	26	1.8	29.04	F
26.91	2	25	1.70	28.80	F	26.91	2	25	1.7	28.8	F
26.86	1	23	1.59	28.54	F	26.86	1	23	1.59	28.54	F
26.64	1	22	1.52	28.36	F	26.64	1	22	1.52	28.36	F
26.52	1	21	1.45	28.17	F	26.52	1	21	1.45	28.17	F
26.5	1	20	1.38	27.97	F	26.5	1	20	1.38	27.97	F
26.27	1	19	1.31	27.77	F	26.27	1	19	1.31	27.77	F
26.09	1	18	1.24	27.55	F	26.09	1	18	1.24	27.55	F
25.91	1	17	1.17	27.32	F	25.91	1	17	1.17	27.32	F
25.64	1	16	1.10	27.09	F	25.64	1	16	1.1	27.09	F
25.36	1	15	1.03	26.83	F	25.36	1	15	1.03	26.83	F
25.13	1	14	0.96	26.57	F	25.13	1	14	0.96	26.57	F
25.09	1	13	0.88	26.28	F	25.09	1	13	0.88	26.28	F
23	1	12	0.81	25.97	F	23	1	12	0.81	25.97	F
22.18	1	11	0.74	25.64	F	22.18	1	11	0.74	25.64	F
21.46	1	10	0.67	25.28	F	21.46	1	10	0.67	25.28	F
19.27	1	9	0.60	24.89	F	19.27	1	9	0.6	24.89	F
18.64	1	8	0.53	24.45	F	18.64	1	8	0.53	24.45	F
17.27	1	7	0.46	23.95	F	17.27	1	7	0.46	23.95	F
8	1	6	0.39	23.39	F	8	1	6	0.39	23.39	F
4.36	1	5	0.32	22.72	F	4.36	1	5	0.32	22.72	F
3.64	1	4	0.25	21.90	F	3.64	1	4	0.25	21.9	F
2.91	2	3	0.14	20.14	F	2.91	2	3	0.14	20.14	F
1.45	1	1	0.04	16.13	F	1.45	1	1	0.04	16.13	F

รูปที่ B20 แสดงผลลัพธ์ช่วงต้น ๆ และช่วงท้ายของตารางเปรียบเทียบผลลัพธ์การคำนวณคะแนนทีปกติจากโปรแกรม

ภาคผนวก C
การสร้างลายเซ็นให้ TScoreCal.jar เพื่อใช้กับ Java Web Start

การที่จะทำให้โปรแกรมที่อยู่ในรูป jar file สามารถทำงานโดยอาศัย Java Web Start นั้น จะต้องทำการสร้างลายเซ็น (sign) ให้แก่ jar file ในที่นี้คือ TScoreCal.jar เสียก่อน

TScoreCal.jar เป็น jar file ที่รวบรวมเอาคลาสต่าง ๆ ที่จำเป็นต้องใช้ในการคำนวณหาคะแนนทีปกติและตัดเกรด มาบีบอัดแล้วจัดเก็บไว้ในที่เดียวกัน ก่อนบีดอัดจะต้องสร้าง manifest file เพื่อให้ JRE รู้ก่อนว่าจะเรียกใช้คลาสใดเป็นคลาสแรกในการเริ่มต้นทำงาน

ข้อความใน manifest file ที่ชื่อว่า myManifest.txt มีเพียงบรรทัดเดียวดังนี้

Main-Class: JApp.TS.TScoreApp

ในการสร้าง TScoreCal.jar ไฟล์ให้ใช้คำสั่งดังนี้

```
C:\JavaMathTools>jar –cvmf myMainifest.txt TScoreCal.jar JApp
MathTools/Common MathTools/Integration MathTools/SimpleStat
```

ขั้นตอนการสร้างลายเซ็น มีดังนี้

1. สร้าง mykey สำหรับการสร้าง self sign certificate เริ่มดังนี้

```
C:\JavaMathTools>keytool  -genkey –keystore mykey –alias me
```

คำสั่งนี้เป็นการให้โปรแกรม keytool สร้าง keystore ชื่อว่า mykey เก็บไว้ในโฟลเดอร์ JavaMathTools ตามด้วย option –alias หมายถึง key ที่สร้างขึ้นนี้จะมีหลากหลายรูปแบบ แต่ที่กำลังจะสร้างให้มีรูปแบบที่ชื่อว่า me

2. รอสักครู่ จะขึ้นหน้าจอให้ใส่ password สำหรับ keystore ดังนี้ ในที่นี้ใส่ physics0 ซึ่งจะไม่แสดงให้เห็นบนจอภาพ

```
Enter keystore password:
Re-enter new password:
```

3. จะมีการถามคำถามอีกหลายบรรทัด เป็นข้อมูลของเราเอง ดังนี้

```
What is your first and last name?
      [Unknown]: Wach R.

What is the name of your organizational unit?
      [Unknown]: physics

What is the name of your organization?
      [Unknown]: RMUTT

What is the name of your City or Locality?
      [Unknown]: Thanyaburi

What is the name of your State or Province?
      [Unknown]: Prathoomtani

What is the two-letter country code for this unit?
      [Unknown]: TH

IS CN=Wach R., OU=physics, O=RMUTT, L=Thanyaburi, ST=Prathoomtani, C=TH
correct?
      [no]: yes
```

4. ขั้นตอนสุดท้ายจะถาม password สำหรับ private key อีกครั้ง

```
Enter key password for <me>
           (RETURN if same as keystore password):
Re-enter new password:
```

Keystore file ที่ชื่อ mykey จะถูกสร้างขึ้นมา เราสามารถเรียกดูโดยใช้ text editor ทั่วไป จะพบว่าเป็นตัวอักษรขอม อ่านไม่รู้เรื่อง เพราะถูกเข้ารหัสเอาไว้

5. ถึงแม้เราจะมี keystore แล้วก็ตามยังไม่สามารถ จะ sign JAR fileได้ จะต้องสร้าง certificate สำหรับการ sign เสียก่อน โดยใช้โปรแกรม keytool ตัวเดิม สามารถสร้าง self signed certificate ได้

```
C:\JavaMathTools>keytools –selfcert –alias me –keystore mykey

Enter keystore password:
```

ถึงขั้นตอนนี้ เรามี certificate พร้อมที่จะ sign JAR file ได้แล้ว

ถ้าต้องการตรวจสอบเพื่อความมั่นใจว่า certificate และ keystore นั้นพร้อมจริง ๆ สามารถใช้คำสั่ง keytool โดยเพิ่ม Option –list ดังนี้

```
C:\JavaMathTools>keytool -list -keystore mykey
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

me, 20 มี.ค. 2551,  PrivateKeyEntry,
Certificate fingerprint (MD5): 2A:50:83:5C:80:80:03:E8:20:63:76:D1:80:94:C7:F1
```

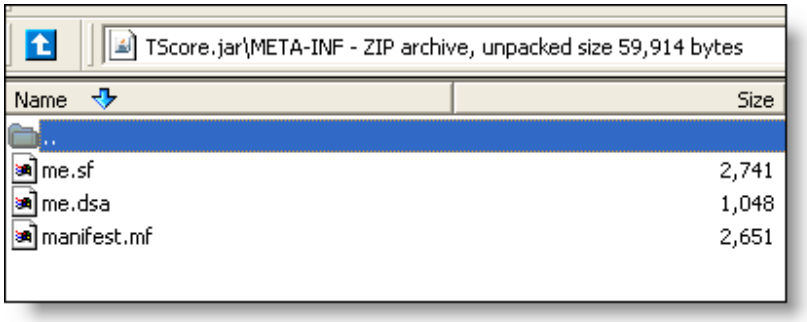
เมื่อใส่ password แล้ว หน้าจอจะแสดง ชนิดของ keystore ผู้สร้าง keystore และ certificate fingerprint

6. การ sign JAR File ทำได้โดยใช้คำสั่ง jarsigner ดังนี้

```
C:\JavaMathTools>jarsigner -keystore mykey TScoreCal.jar me
Enter Passphrase for keystore:

Warning:
The signer certificate has expired.
```

ถึงขั้นตอนนี้แสดงว่าเรา sign JAR file ได้สำเร็จแล้ว

เมื่อเราเข้าไปดูใน JAR file พบว่า จะมีไฟล์เกิดขึ้นใหม่ 2 ไฟล์ คือ me.dsa (digital signature) และ me.sf เฉพาะ me.sf เท่านั้นที่สามารถอ่านได้จาก text editor



```
Signature-Version: 1.0
SHA1-Digest-Manifest-Main-Attributes: c1QfBTwcMtlZ5gHE8SWlwWcRgWM=
Created-By: 1.6.0_04 (Sun Microsystems Inc.)
SHA1-Digest-Manifest: 9sCNsYgsszamNvnDRBuSuqE6kgI=

Name: MathTools/SimpleStat/SimpleStat.class
SHA1-Digest: /Wnv0ugZjlGDqX2Tt1dogfcNxQY=

Name: JApp/TS/GradeOptionPanel.class
SHA1-Digest: MrQ4ilUsrvHFdtjvVSb6elPygyU=

Name: MathTools/Common/PhysicsConstants.class
SHA1-Digest: dZT/csRs70GNi+CjwfrY5/asqGw=

Name: MathTools/Integration/Simpson1_3Integration.class
SHA1-Digest: cxVP2HXydRCa6kUe7Mqpwa4qeG8=

Name: JApp/TS/TScoreApp.class
SHA1-Digest: sLx0WiDo99qXYmMSJ8e2ebmphOg=

Name: MathTools/SimpleStat/ScoreTextArea.class
SHA1-Digest: oPKYFHy5HgDoQQX+OHJTDc6bw/s=

Name: MathTools/Integration/Integrator.class
SHA1-Digest: GI20eehuYA9u8BPuwWvak+hNX+o=
```

และ เมื่อเปิดไฟล์ manifest.mf พบว่าจะมีข้อความอีก หลายบรรทัดเพิ่มขึ้นมา

```
Manifest-Version: 1.0
Created-By: 1.6.0_04 (Sun Microsystems Inc.)
Main-Class: JApp.TS.TScoreApp

Name: MathTools/SimpleStat/SimpleStat.class
SHA1-Digest: SuBjyTFRQ70IB/ivr4Fv8limJP4=

Name: JApp/TS/GradeOptionPanel.class
SHA1-Digest: DKTA67T0CDGWkbScEVe9vy7WO/8=

Name: MathTools/Common/PhysicsConstants.class
SHA1-Digest: O7hrGS5S3zR0qJLPgKm5s9UuyAA=

Name: MathTools/Integration/Simpson1_3Integration.class
SHA1-Digest: bOtaFWH2MY+wkueiDQmlxqsmv2Y=

Name: JApp/TS/TScoreApp.class
SHA1-Digest: uBHOifHIUXy4In/MTDfsnhs/p4s=

Name: MathTools/SimpleStat/ScoreTextArea.class
SHA1-Digest: ANPhrxGh7sjEN+G0AgpNHUR5wvc=

Name: MathTools/SimpleStat/TScoreConstants.class
SHA1-Digest: are3LShTK7/fP8j+g03bGJIGUuw=

Name: MathTools/Integration/Integrator.class
```

7. ถ้าต้องการตรวจสอบ ว่า jarsigner สามารถ sign jar file ได้ถูกต้องไหม เราสามารถใช้คำสั่ง jarsigner ตรวจสอบดูได้ ดังนี้

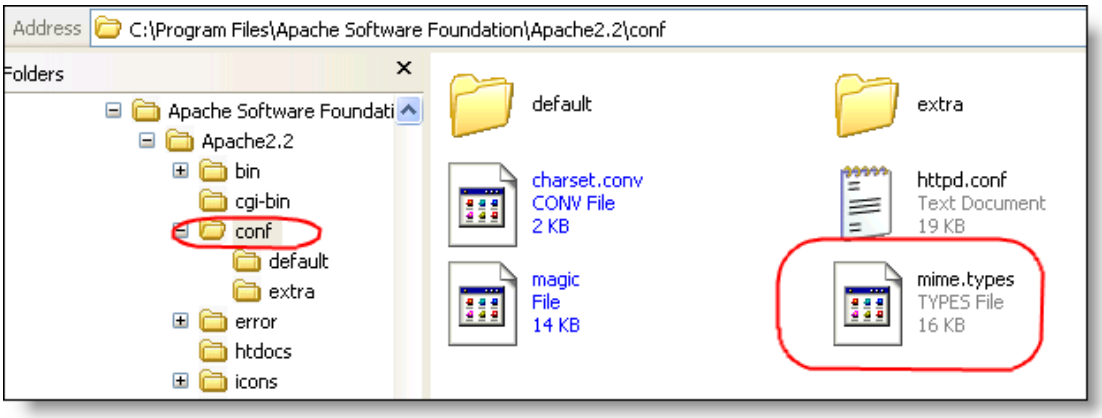
```
C:\JavaMathTools>jarsigner -verbose -verify -certs TScoreCal.jar
```

จะแสดง Output ได้ผลลัพธ์ดังนี้

```
        2651 Fri Jun 20 21:01:44 GMT-12:00 2008 META-INF/MANIFEST.MF
        2741 Fri Jun 20 21:01:44 GMT-12:00 2008 META-INF/ME.SF
        1048 Fri Jun 20 21:01:44 GMT-12:00 2008 META-INF/ME.DSA
           0 Thu Jun 19 22:37:12 GMT-12:00 2008 META-INF/
           0 Wed Jun 18 10:30:02 GMT-12:00 2008 JApp/
           0 Wed Jun 18 10:31:04 GMT-12:00 2008 JApp/TS/
sm      2061 Wed Jun 18 11:50:52 GMT-12:00 2008 JApp/TS/AboutPanel.class

      X.509, CN=Wach R., OU=physics, O=RMUTT, L=Thanyaburi, ST=Prathoomtani, C=T
H
      [certificate expired on 18/9/2494, 20:52 น.]

sm       291 Thu Jun 19 22:17:10 GMT-12:00 2008 MathTools/SimpleStat/TScoreConst
ants.class

      X.509, CN=Wach R., OU=physics, O=RMUTT, L=Thanyaburi, ST=Prathoomtani, C=T
H
      [certificate expired on 18/9/2494, 20:52 น.]


  s = signature was verified
  m = entry is listed in manifest
  k = at least one certificate was found in keystore
  i = at least one certificate was found in identity scope

jar verified.

Warning:
This jar contains entries whose signer certificate has expired.
```

8. ตั้งค่าระบบของ Web server (หรือที่เรียกว่า configure web server) โดยเพิ่ม MIME type ซึ่งจะเป็นตัวบอกให้ Web server รู้ว่า JNLP file นั้นจะต้องใช้คู่กับ java

กรณีที่ Web server เป็น Apache สำหรับระบบปฏิบัติการวินโดว์ เข้าไปที่โฟลเดอร์ /conf/mime.types ระบบปฏิบัติการลินุกซ์ อาจเป็น /etc/mime.types



เปิดไฟล์ mime.types โดยใช้ text editor เพิ่มเติมข้อความต่อไปนี้

Application/x–java-jnlp

จากนั้นให้ restart server เพื่อให้ server รับรู้ค่าที่ตั้งขึ้นใหม่

กรณีที่ Web server เป็น IIS (Internet Information Services)

- เรียกใช้ administrator tools แล้วคลิกเมาส์ที่ IIS Management Console เมื่อหน้าต่างของ IIS Management ปรากฏขึ้น ให้คลิกเมาส์ปุ่มขวาที่ชื่อ Server ที่panel ด้านซ้ายมือ เลือก Properties จาก drop-down menu

- ที่หัวข้อ MIME Map คลิกที่ปุ่ม Edit

- หน้าต่าง ที่มีชื่อว่า File Type จะปรากฏขึ้นมา คลิกที่ New Type

- ตรงข้อความ File Type box พิมพ์คำว่า jnlp

- ตรงข้อความ Content Type (MIME) box ให้พิมพ์ข้อความ Application/x-java-jnlp-file jnlp

- คลิก OK เพื่อปิด File Type box คลิก OK อีกครั้งเพื่อปิดหน้าต่าง File Type คลิก OK ที่หน้าต่าง Properties และ save the changes

- Restart server เพื่อให้รับรู้ค่าที่ตั้งขึ้นใหม่

9. นำไฟล์ต่อไปนี้ไปติดตั้งที่ Web Server มีไฟล์หลักอยู่ 3 ไฟล์ ดังนี้

TScoreCal.jnlp เป็นไฟล์ที่ใช้บอกคุณลักษณะต่าง ๆ เพื่อให้ Java Web Start รู้ว่าจะเริ่มต้นให้โปรแกรมทำงานได้อย่างไร

index.html เป็น html file ที่มีลิงค์ ไปยัง TScoreCal.jnlp

TScoreCal.jar JAR file ที่ได้รับการ sign แล้ว สามาถทำงานได้โดยผ่านทาง Java Web Start

TScoreCal.jnlp เป็นไฟล์ที่เขียนด้วย XML เป็นไฟล์ที่ใช้บอกให้ Java Web Start รู้ว่าจะรันโปรแกรมนี้ในลักษณะอาการอย่างไรบ้าง ข้อความในเอกสารนี้มีดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
<!–JNLP file for Normalized T Score Calculation -->
<jnlp
  spec="1.0+"
  codebase="http://203.158.100.140/TScore"
  href="TScoreCal.jnlp">

  <information>
    <title>Normalize T Score Calculation.</title>
    <vendor>RMUTT Physics</vendor>

    <homepage href="http://203.158.100.140/TScore/index.htm"/>

    <description>The utility for calculating T Score</description>
    <description kind="short">Normalized T Score Calculation via the
capabilities of JAVA WebStart.</description>

    <offline-allowed/>
  </information>

  <security>
     <all-permissions/>
  </security>

  <resources>
    <j2se version="1.6"/>
    <jar href="TScoreCal.jar"/>
  </resources>

  <application-desc main-class="JApp.TS.TScoreApp"/>
```

```
</jnlp>
```

อธิบายความหมายของแต่ละประโยค ดังนี้

```
<?xml version="1.0" encoding="utf-8"?>
```

เป็นระบุคุณลักษณะเฉพาะ (specification) ของ ภาษาXML เพื่อให้ตัวแปลภาษาหรือ parser สามารถรับรู้และแปลได้ถูกต้อง

```
<!—JNLP file for Normalized T Score Calculation -->
```

เป็น comment ของไฟล์ เพื่อบอกให้รู้ว่าไฟล์ JNLP นี้ใช้กับโปรแกรมใด

```
<jnlp
  spec="1.0+"
  codebase="http://203.158.100.140/TScore"
  href="TScoreCal.jnlp">
```

จัดเป็น tag ที่สำคัญหรือต้องมีของ jnlp file ทุกไฟล์ บอกถึงโปรโตคอลของ Java Web Start ว่าจะใช้รุ่นใด ในที่นี้ใช้ 1.0+ บอกถึงไฟล์นั้นอยู่ที่โฟลเดอร์ใด และชื่อไฟล์ jnlp ในที่นี้คือ TScoreCal.jnlp

```
<information>
    <title>Normalize T Score Calculation.</title>
    <vendor>RMUTT Physics</vendor>

    <homepage href="http://203.158.100.140/TScore/index.htm"/>

    <description>The utility for calculating T Score</description>
    <description kind="short">Normalized T Score Calculation via the
capabilities of JAVA WebStart.</description>

    <offline-allowed/>
  </information>
```

ส่วนนี้จะเป็นส่วนที่ใช้บอกข้อมูลทั่วไปของโปรแกรมให้ผู้ใช้ได้ทราบในระหว่างการติดตั้งโปรแกรม ข้อความใน <title> จะกลายเป็นชื่อ icon บน desktop ของผู้ใช้ homepage ที่ใช้ต้องเปลี่ยนให้สอดคล้องกับชื่อ homepage ที่ติดตั้ง ถ้าผู้ใช้จำลองเครื่องคอมพิวเตอร์ให้เป็น Web server ต้องเปลี่ยนเลข IP ให้เป็น 127.0.0.1

```
  <security>
     <all-permissions/>
  </security>
```

ส่วนนี้จะบอกว่า สามารถเข้าถึงเครื่องคอมพิวเตอร์ของผู้ใช้ ได้ทุกรูปแบบ

```
<resources>
    <j2se version="1.6+"/>
    <jar href="TScoreCal.jar"/>
  </resources>
```

บอกถึงรุ่นของจาวาที่สามารถใช้ได้กับโปรแกรมนี้ ในที่นี้ให้ใช้ JRE ตั้งแต่ 1.6 ขึ้นไป jar file ที่เรียกใช้งานคือ TScoreCal.jar

```
  <application-desc main-class="JApp.TS.TScoreApp"/>
```

ระบุถึงคลาสเริ่มต้น ที่จะให้โปรแกรมนี้ทำงาน

index.html เป็นไฟล์เอกสารที่เขียนโดยใช้ภาษา html เมื่อผู้ใช้เชื่อมต่ออินเตอร์เน็ตเข้ามาที่ website นี้ ไฟล์ index.html จะเป็นไฟล์เริ่มต้นที่แสดงผลในบราวเซอร์ของผู้ใช้ ข้อความใน index.html จะต้องมี tag ต่อไปนี้ อยู่ในส่วนหนึ่งส่วนใดของเอกสาร จึงจะสามารถเรียกใช้งาน TScoreCal.jar ได้

Run โปรแกรม Normalized T Score คลิกที่นี่

TScoreCal.jar เป็น jar file ที่รวบรวมเอาคลาสต่าง ๆ ที่จำเป็นต้องใช้ในการคำนวณหาคะแนนทีปกติ และตัดเกรด มาบีบอัดแล้วจัดเก็บไว้ในที่เดียวกัน และมีลายเซ็นเรียบร้อยแล้ว จึงจะใช้งานใน Java Web Start ได้

This page left blank intentionally.

ภาคผนวก D
คู่มือการใช้งานโปรแกรม TScoreCal

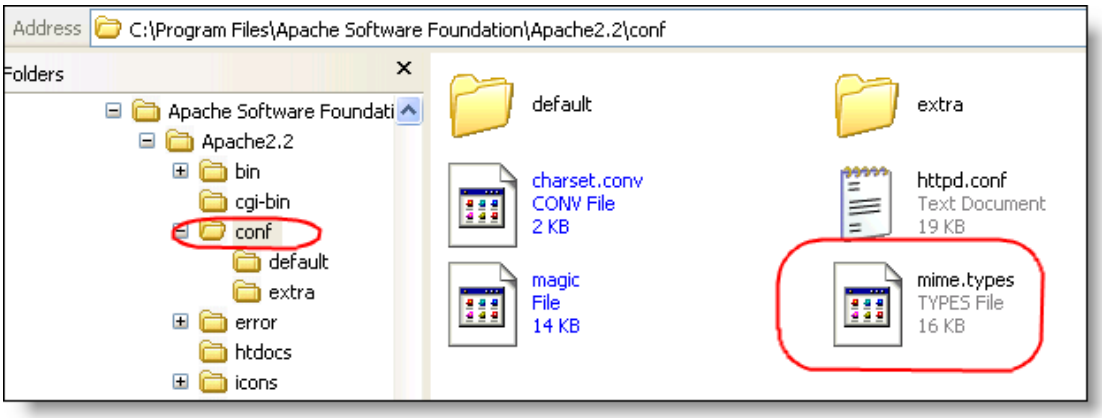
โปรแกรม Normalized T Score Calculation หรือเรียกสั้น ๆ ว่า TScoreCal เป็นโปรแกรมแปลงคะแนนดิบให้เป็นคะแนนที ปกติ และกำหนดระดับคะแนนหรือตัดเกรด อำนวยความสะดวกทำให้ผู้สอนสามารถประเมินผลการเรียนการสอนได้รวดเร็วยิ่งขึ้น โปรแกรมนี้รองรับจำนวนนักศึกษาได้ 2000 คน โปรแกรมนี้จะถูกเก็บไว้ใน Web Server ผู้สอนหรือผู้ต้องการใช้งานสามารถเรียกใช้โดยการเชื่อมต่ออินเตอร์เนตไปยัง web site นั้น โปรแกรมจะถูกดาวน์โหลดมายังเครื่องคอมพิวเตอร์ของผู้ใช้และประมวลผลทันที โดยไม่ต้องทำการติดตั้งให้เปลืองเนื้อที่ในฮาร์ดดิสก์ แต่ถ้าผู้ใช้งานต้องการเก็บโปรแกรมนี้ไว้ใช้งานภายหลังจากเลิกเชื่อมต่ออินเตอร์เน็ตแล้ว ก็ทำได้โดยการ save โปรแกรมนี้เก็บไว้ในคอมพิวเตอร์ของตนได้เช่นกัน

ตอนที่ 1 สำหรับ Web Master

ในตอนที่ 1 นี้สำหรับ Web Master ที่ต้องการนำโปรแกรมนี้ไปไว้ใน Web Server สามารถทำได้ตามลำดับขั้นตอนดังนี้

1. นำไฟล์ทั้งหมดภายใต้โฟลเดอร์ TScore ไปเก็บไว้ใน root folder ของ web page โดยทั่วไปมักจะอยู่ที่ C:\WWW หรือ ถ้าใช้ IIS เป็น Web Server จะอยู่ที่ C:\InetPub
2. กำหนดค่า MIME Type ให้แก่โปรแกรม Web Server

2.1 กรณีที่ Web server เป็น Apache สำหรับระบบปฏิบัติการวินโดว์ เข้าไปที่โฟลเดอร์ /conf/mime.types ระบบปฏิบัติการลินุกซ์ อาจเป็น /etc/mime.types



รูปที่ D1 แสดงตำแหน่งที่เก็บไฟล์ชื่อ mine.types

เปิดไฟล์ mime.types โดยใช้ text editor เพิ่มเติมข้อความต่อไปนี้

Application/x–java-jnlp

จากนั้นให้ restart server เพื่อให้ server รับรู้ค่าที่ตั้งขึ้นใหม่

2.2 กรณีที่ Web server เป็น IIS (Internet Information Services)

- เรียกใช้ administrator tools แล้วคลิกเมาส์ที่ IIS Management Console เมื่อหน้าต่างของ IIS Management ปรากฏขึ้น ให้คลิกเมาส์ปุ่มขวาที่ชื่อ Server ที่panel ด้านซ้ายมือ เลือก Properties จาก drop-down menu

- ที่หัวข้อ MIME Map คลิกที่ปุ่ม Edit

- หน้าต่าง ที่มีชื่อว่า File Type จะปรากฏขึ้นมา คลิกที่ New Type

- ตรงกล่องข้อความ File Type box พิมพ์คำว่า jnlp

- ตรงกล่องข้อความ Content Type (MIME) box ให้พิมพ์ข้อความ Application/x-java-jnlp-file jnlp

- คลิก OK เพื่อปิด File Type box คลิก OK อีกครั้งเพื่อปิดหน้าต่าง File Type คลิก OK ที่หน้าต่าง Properties และ save the changes

- Restart server เพื่อให้รับรู้ค่าที่ตั้งขึ้นใหม่

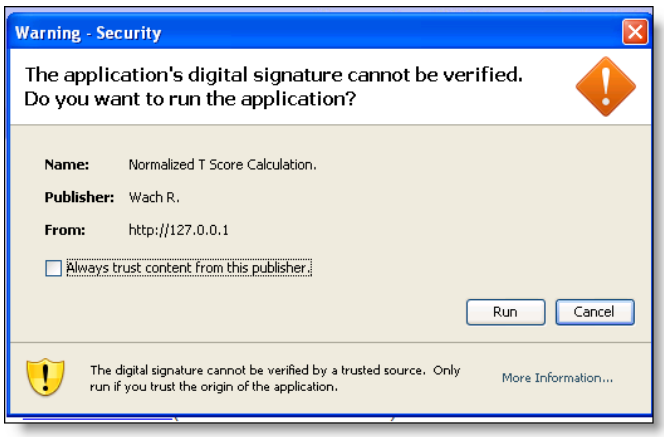
ตอนที่ 2 สำหรับผู้ใช้ทั่วไป (End User)

ต้องการใช้งานโปรแกรมนี้ ไปที่เว็บไซต์ฟิสิกส์ราชมงคล (http://203.158.100.140/Tscore/) จะพบกับหน้าแรกของโฮมเพจ หน้าตาดังนี้

รูปที่ D2 หน้าแรกของโฮมเพจการคำนวณคะแนนทีปกติ

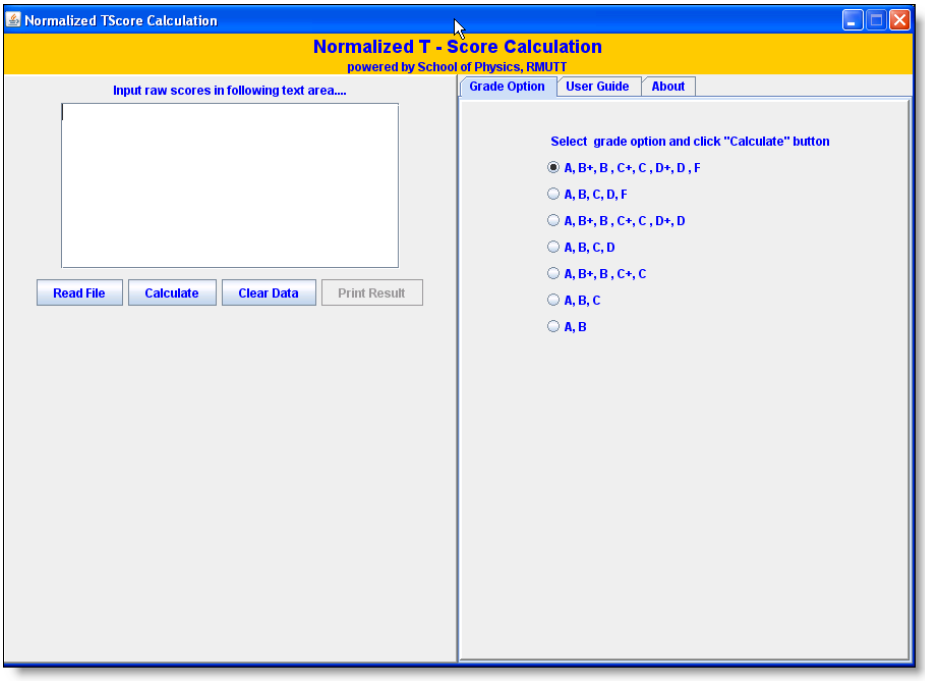
ต้องการใช้งานโปรแกรม คลิกที่ข้อความ “Run โปรแกรม Normalized T Score คลิกที่นี้”

โปรแกรมจะถูกดาวน์โหลดเข้ามาทำงานที่เครื่องคอมพิวเตอร์ของผู้ใช้ทันที จะมีหน้าต่างโผล่ขึ้นมาดังรูป ถามว่า ต้องการจะรันโปรแกรมนี้หรือไม่



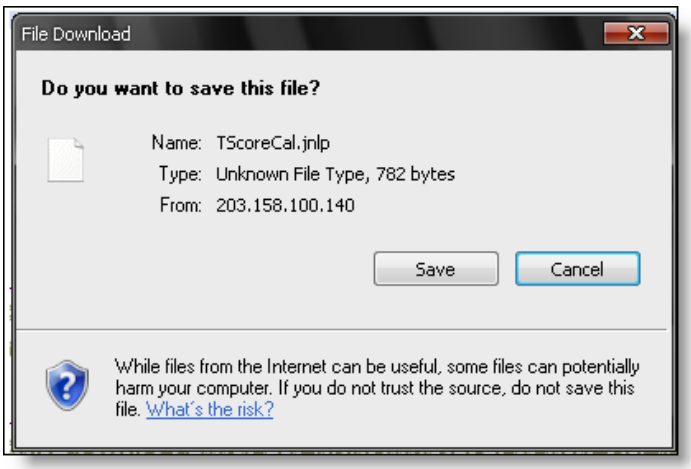
รูปที่ D3 หน้าต่างแสดงข้อความยืนยันการ Run โปรแกรม

ให้คลิกที่ปุ่ม Run ที่ด้านล่างของหน้าต่าง รอสักครู่หนึ่ง หน้าตาของโปรแกรมจะปรากฏให้เห็น



รูปที่ D4 หน้าตาของโปรแกรม TScoreCal

ถ้าคอมพิวเตอร์เครื่องใด มีหน้าต่าง File Download แจ้งว่าต้องการ save ไฟล์นี้หรือไม่ดังรูปต่อไปนี้ แสดงว่าโปรแกรมนี้ยังไม่สามารถรันในเครื่องคอมพิวเตอร์เครื่องนี้ได้ ผู้ใช้งานจะต้องไปดาวน์โหลด JRE (Java Runtime Environment) หรือสภาพแวดล้อมที่ทำให้โปรแกรมที่เขียนด้วยภาษาจาวาทำงานได้ภายในเครื่องนี้เสียก่อน โดยคลิกที่ลิงค์ ตรงข้อความที่เขียนไว้ว่า Download JRE 1.6 ลิงค์นี้จะนำไปสู่เว็บไซต์ www.java.com ให้ดาวน์โหลดมาติดตั้งให้เรียบร้อย แล้วจึงคลิกที่ลิงค์ “Run โปรแกรม Normalized T Score คลิกที่นี้” อีกครั้ง



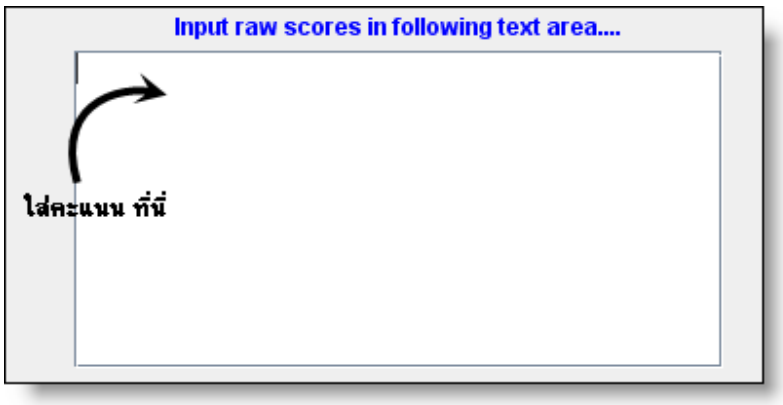
รูปที่ D5 หน้าต่างแสดงให้ save ไฟล์ กรณีที่ไม่สามารถ Run โปรแกรมได้

โปรแกรม TScoreCal นี้ทำงานนอกบราวเซอร์ สามารถอ่านไฟล์และส่งผลลัพธ์ไปยังเครื่องพิมพ์ได้

ขั้นตอนการใช้งานโปรแกรม

ขั้นตอนที่ 1 ป้อนข้อมูล เริ่มต้นด้วยการป้อนคะแนนดิบเข้าไปตรงบริเวณ Text Area ซึ่งทำได้ 2 วิธี คือ

1. ป้อนคะแนนดิบผ่านทางแป้นพิมพ์โดยตรง

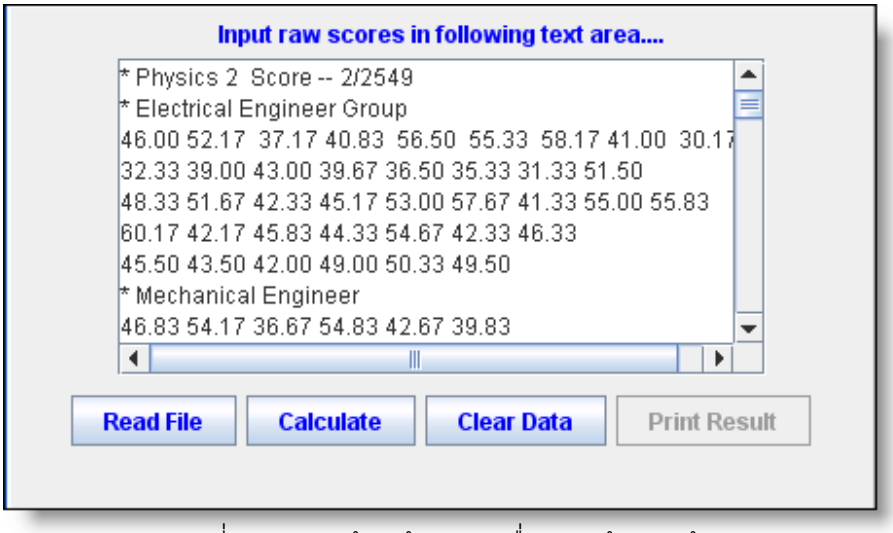


รูปที่ D6 แสดงกรอบป้อนข้อความสำหรับพิมพ์คะแนน

หลักเกณฑ์ในการพิมพ์คะแนนมีดังนี้

- คะแนนจะเป็นจำนวนเต็มหรือจำนวนที่มีทศนิยมก็ได้ แยกคะแนนแต่ละคะแนนด้วยช่องว่า หรือเคาะ tab key ห้ามใช้เครื่องหมายจุลภาค หรือตัวอักขระใด ๆ
- บรรทัดหนึ่ง ๆ จะพิมพ์คะแนนกี่ค่าก็ได้ ต้องการขึ้นบรรทัดใหม่ให้กดปุ่ม Enter
- ต้องการเขียนหมายเหตุ หรือบันทึกช่วยจำ ให้ใส่ * หน้าบรรทัดนั้น โปรแกรมจะไม่นำบรรทัดที่มีเครื่องหมาย * ไปประมวลผล
- สามาถใช้โปรแกรม editor เช่น EditPlus NotePad พิมพ์คะแนนเตรียมไว้ก่อนแล้วจึงใช้คำสั่ง copy และ paste ไปวางไว้ใน Text area นี้ได้เช่นกัน

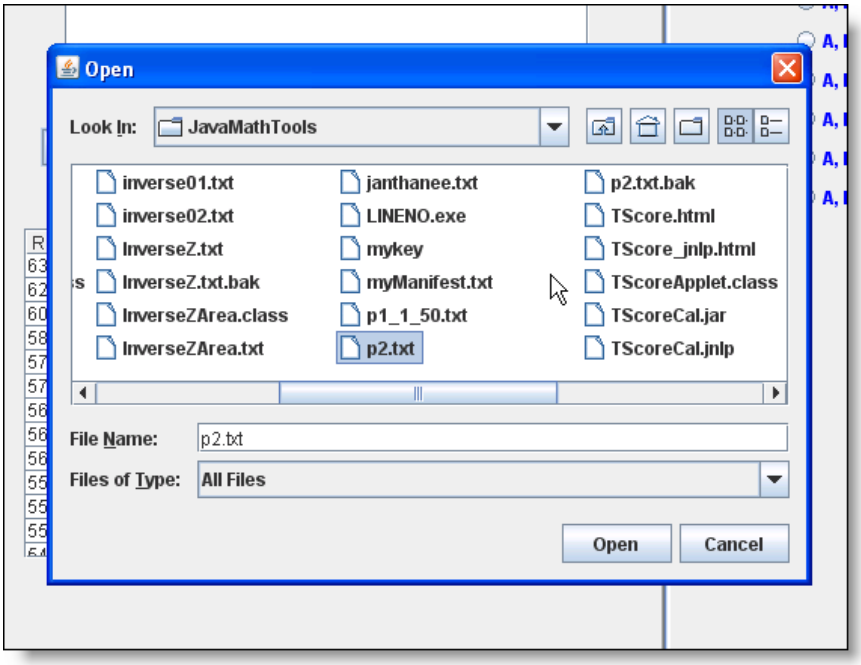
ตัวอย่าง การพิมพ์คะแนนลงไปใน Text area



รูปที่ D7 กรอบป้อนข้อความเมื่อกรอกข้อมูลแล้ว

2. อ่านคะแนนจาก text file

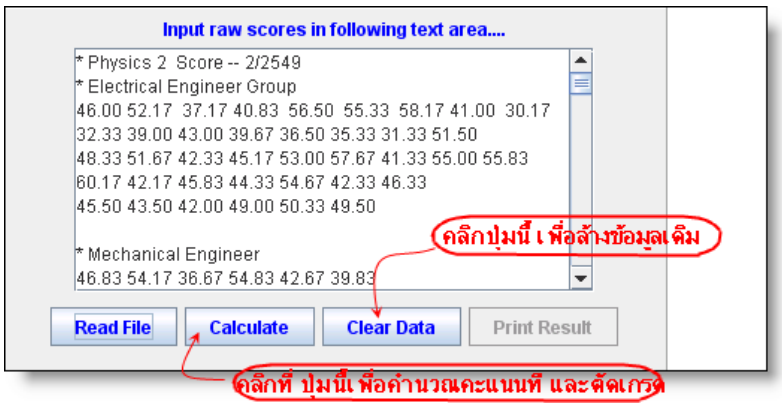
กรณีนี้เหมาะกับข้อมูลที่มีเป็นจำนวนมาก นำคะแนนดิบไปพิมพ์ด้วย Text editor แล้ว save เป็น text file เก็บไว้ หลักเกณฑ์ในการพิมพ์คะแนน ใช้หลักเกณฑ์เดียวกับที่กล่าวมาแล้วในข้อ 1 การนำคะแนนจาก text file เข้ามาใน text area ทำได้โดยคลิกที่ปุ่ม Read file เลือกไฟล์ที่จัดเก็บข้อมูลไว้ คลิก ปุ่ม Open จะได้ข้อมูลนำมาเก็บไว้ใน text area



รูปที่ D8 การเปิดไฟล์เพื่อนำข้อมูลเข้ามาในกลองข้อความ

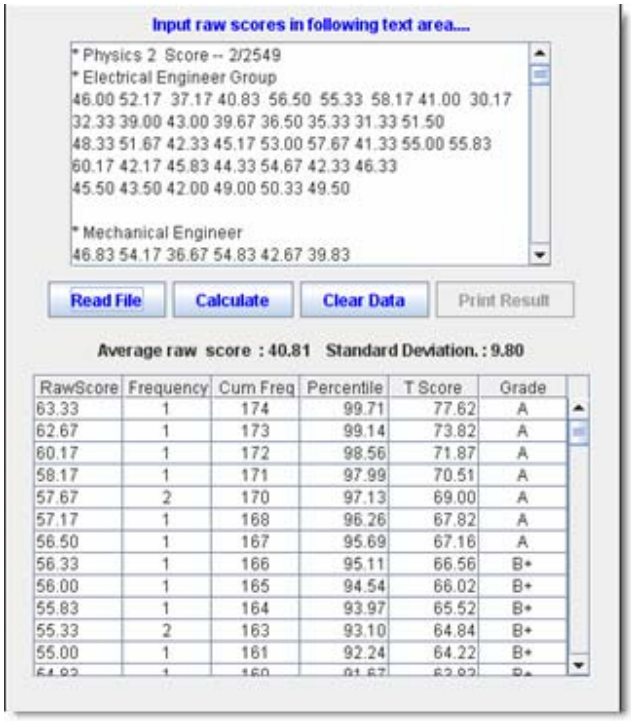
<u>ขั้นตอนที่ 2</u> คำนวณคะแนนทีปกติ และกำหนดเกรด

เมื่อคะแนนดิบถูกป้อนเข้าสู่บริเวณ Text area เรียบร้อยแล้ว ให้คลิกปุ่ม Calculate เพื่อทำการประมวลผล



รูปที่ D9 คลิกปุ่มต่าง ๆ ให้โปรแกรมทำงาน

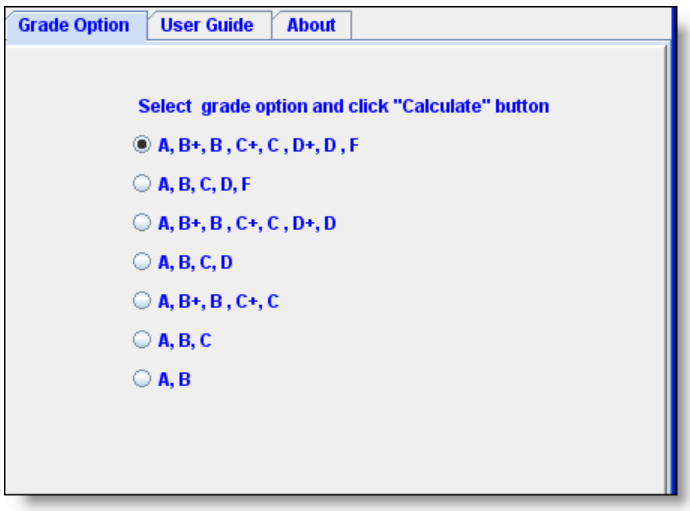
ผลลัพธ์ที่ได้จากการคำนวณได้แก่ ความถี่ ความถี่สะสม Percentile คะแนนทีปกติ รวมทั้งเกรดที่ควรได้รับ จะปรากฏเป็นตารางใต้บริเวณ Text area



RawScore	Frequency	Cum Freq	Percentile	T Score	Grade
63.33	1	174	99.71	77.62	A
62.67	1	173	99.14	73.82	A
60.17	1	172	98.56	71.87	A
58.17	1	171	97.99	70.51	A
57.67	2	170	97.13	69.00	A
57.17	1	168	96.26	67.82	A
56.50	1	167	95.69	67.16	A
56.33	1	166	95.11	66.56	B+
56.00	1	165	94.54	66.02	B+
55.83	1	164	93.97	65.52	B+
55.33	2	163	93.10	64.84	B+
55.00	1	161	92.24	64.22	B+

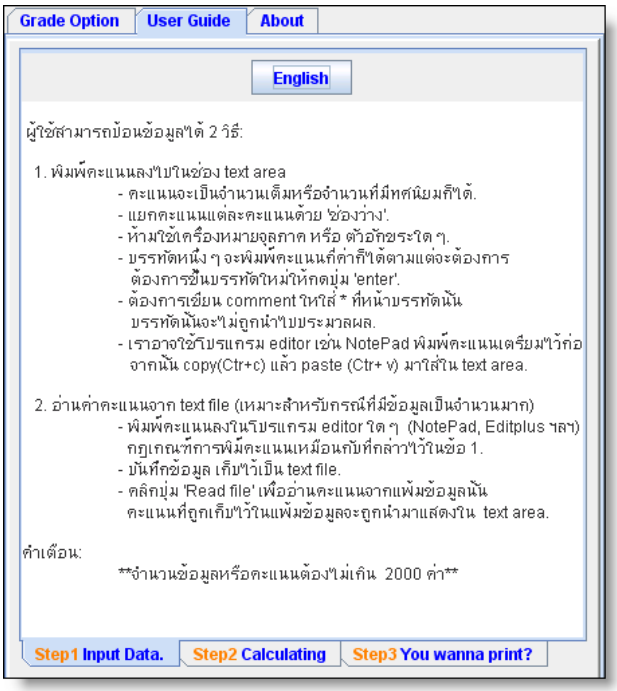
รูปที่ D10 ผลลัพธ์ของการคำนวณจะแสดงในรูปตาราง

ต้องการเปลี่ยนจำนวนเกรดที่ต้องการตัด ให้คลิกปุ่มใน panel ด้านขวามือ แผ่นแท็บ ที่มีชื่อว่า Grade Option ให้ตรงกับกับจำนวนเกรดที่ต้องการ แล้วคลิกที่ปุ่ม Calculate อีกครั้งเพื่อให้โปรแกรมประมวลผลใหม่



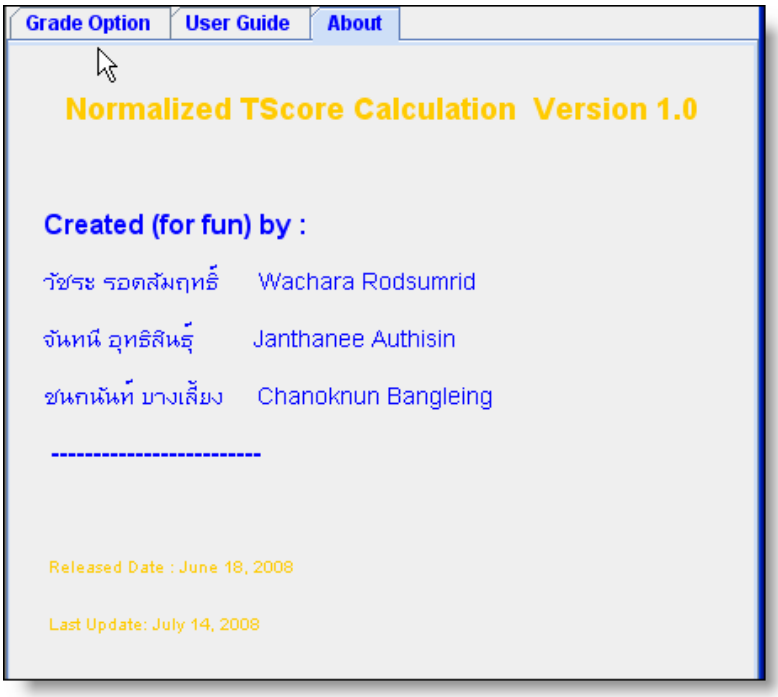
รูปที่ D11 หน้าจอแสดงการเลือกช่วงการตัดเกรด

แผ่นแท็บที่มีหัวเรื่อง User Guide เป็นการแนะนำการใช้งาน โปรแกรมนี้ ซึ่งมีข้อความคล้ายกับคู่มือที่กำลังอ่านอยู่ขณะนี้



รูปที่ D12 แสดงแท็บ User Guide

แผ่นแท็บที่มีหัวเรื่อง About จะแสดงชื่อผู้พัฒนาโปรแกรมนี้ รุ่นของโปรแกรม และ วันที่เริ่มใช้โปรแกรมนี้ และวันที่ปรับปรุงครั้งล่าสุด



รูปที่ D13 แท็บ About บอกชื่อผู้พัฒนาโปรแกรม

<u>ขั้นตอนที่ 3</u> พิมพ์ผลลัพธ์ของโปรแกรมทางเครื่องพิมพ์

ต้องการเก็บผลลัพธ์ไว้อ้างอิงในภายหน้า สามารถทำได้โดยคลิกที่ปุ่ม Print result ผลลัพธ์ที่ปรากฏในตาราง รวมทั้งค่าเฉลี่ย และส่วนเบี่ยงเบนมาตรฐาน จะถูกพิมพ์ออกมาทางเครื่องพิมพ์ โปรแกรมนี้ตั้งค่าขนาดของกระดาษที่ใช้พิมพ์ไว้เป็น ขนาด A4

เมื่อโปรแกรมนี้เริ่มต้นทำงาน ปุ่ม Print result จะมีสีจาง ๆ ไม่สามารถคลิกใช้งานได้ ก็ต่อเมื่อได้กดปุ่ม Calculate ปุ่มนี้จึงจะมีสีเข้ม และสามารถคลิกใช้งานได้

ตัวอย่างเอกสารที่พิมพ์ได้ ดังรูปต่อไปนี้

Normalized T Score Calculation Report

Data from : C:\JavaMathTools\p2.txt

Printed Date : Mon Jul 14 19:29:19 GMT-12:00 2008

*** Average raw score : 40.81 Standard Deviation. : 9.80

RawScore	Frequency	Cum Freq	Percentile	T Score	Grade
63.33	1	174	99.71	77.62	A
62.67	1	173	99.14	73.82	A
60.17	1	172	98.56	71.87	A
58.17	1	171	97.99	70.51	A
57.67	2	170	97.13	69.00	A
57.17	1	168	96.26	67.82	A
56.50	1	167	95.69	67.16	A
56.33	1	166	95.11	66.56	B+
56.00	1	165	94.54	66.02	B+
55.83	1	164	93.97	65.52	B+
55.33	2	163	93.10	64.84	B+
55.00	1	161	92.24	64.22	B+
54.83	1	160	91.67	63.83	B+
54.67	1	159	91.09	63.46	B+
54.17	1	158	90.52	63.12	B+
53.00	1	157	89.94	62.78	B+
52.33	1	156	89.37	62.46	B+
52.17	2	155	88.51	62.01	B+
51.83	3	153	87.07	61.30	B+
51.67	3	150	85.34	60.51	B
51.50	2	147	83.91	59.91	B
50.67	2	145	82.76	59.45	B
50.33	1	143	81.90	59.11	B
50.00	1	142	81.32	58.90	B
49.50	3	141	80.17	58.48	B
49.00	1	138	79.02	58.07	B
48.33	1	137	78.45	57.87	B
48.00	1	136	77.87	57.68	B
47.83	1	135	77.30	57.49	B
47.33	1	134	76.72	57.30	B
46.83	1	133	76.15	57.11	B
46.33	3	132	75.00	56.74	B
46.00	1	129	73.85	56.39	B
45.83	1	128	73.28	56.21	B
45.50	1	127	72.70	56.04	B
45.17	2	126	71.84	55.78	B
45.00	2	124	70.69	55.44	C+
44.83	3	122	69.25	55.03	C+

รูปที่ D14 บางส่วนของผลลัพธ์ที่ได้จากเครื่องพิมพ์

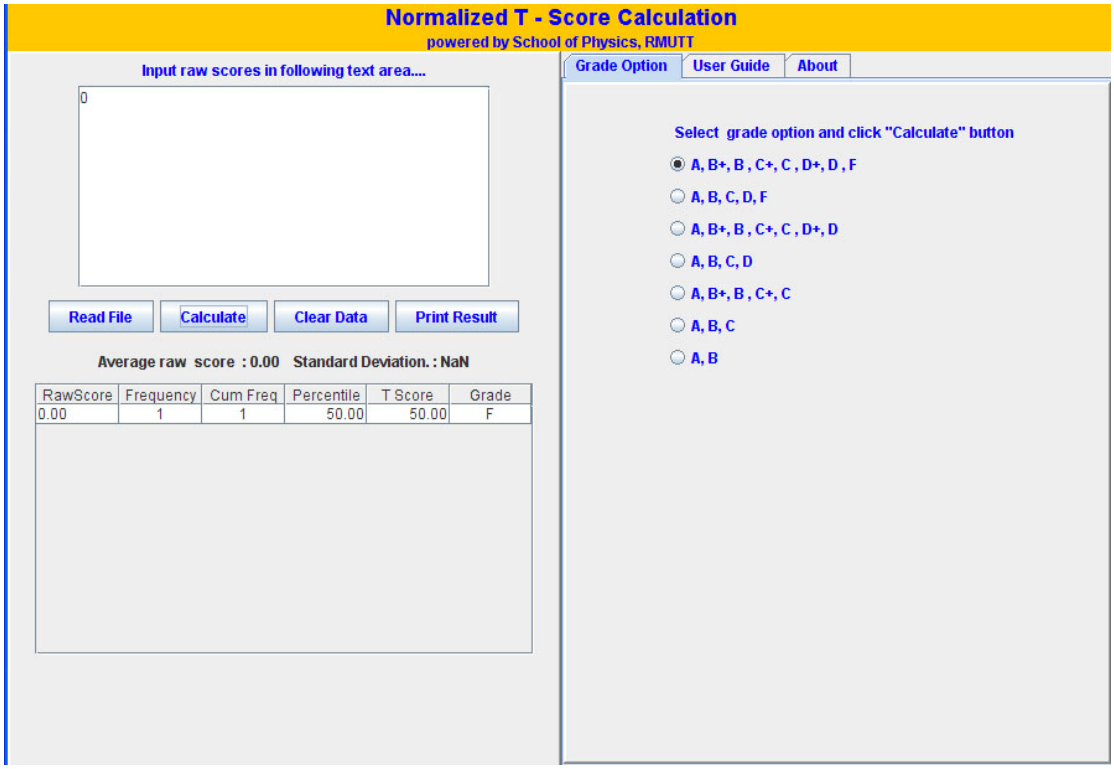
ขั้นตอนการใช้งานมีอยู่เพียง 3 ขั้นตอนเท่านั้น ต้องการรายละเอียดอื่น ๆ เกี่ยวกับคะแนนทีปกติ สามารถอ่านได้จากบทความในเว็บไซต์ http://203.158.100.140/TScore/

ภาคผนวก D
การทดสอบโปรแกรม TScoreCal

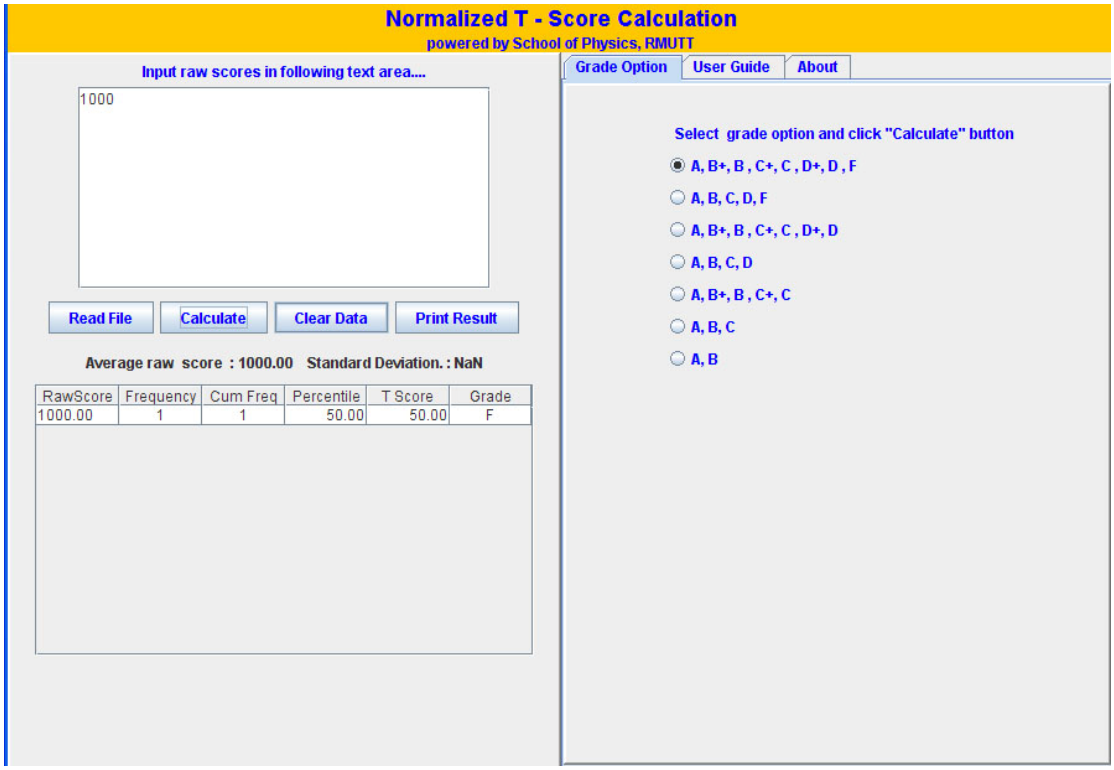
ต่อไปนี้เป็นผลการทดสอบโปรแกรม TScoreCal ในรูปแบบต่าง ๆ เพื่อหาข้อผิดพลาดของโปรแกรม

1. การทดสอบโดยให้จำนวน 1 ตัว

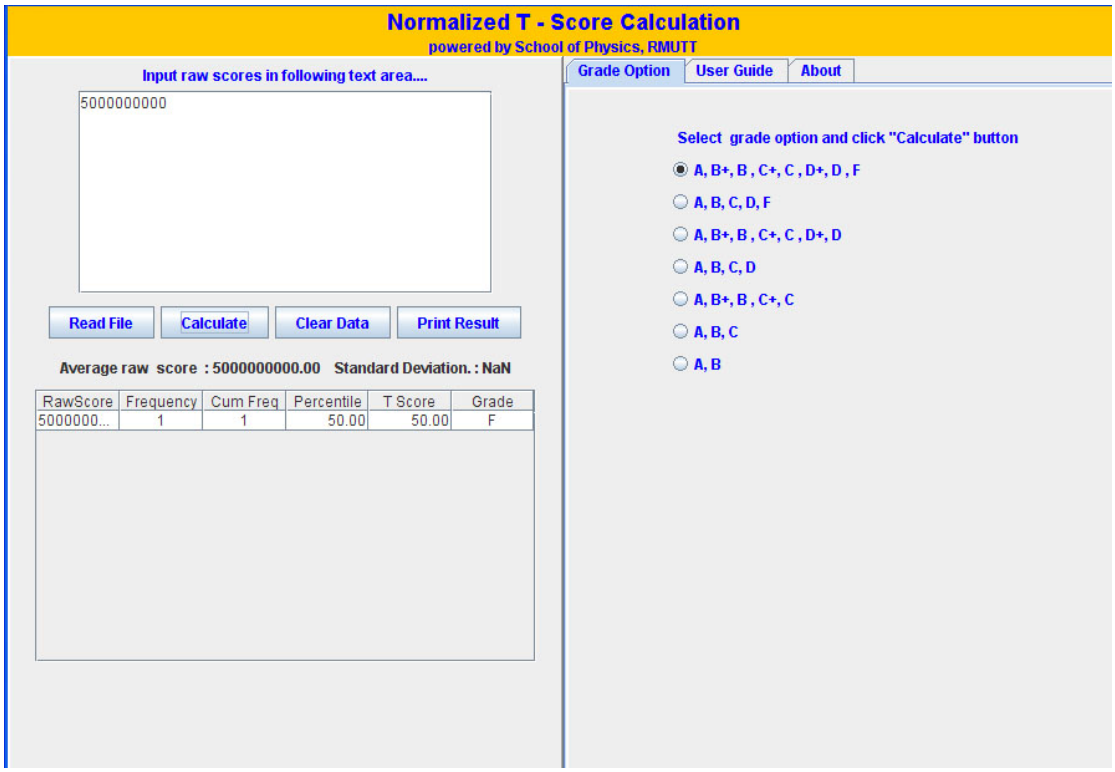
1.1 ให้ค่าคะแนนเป็น 0



1.2 ให้คะแนน 1,000



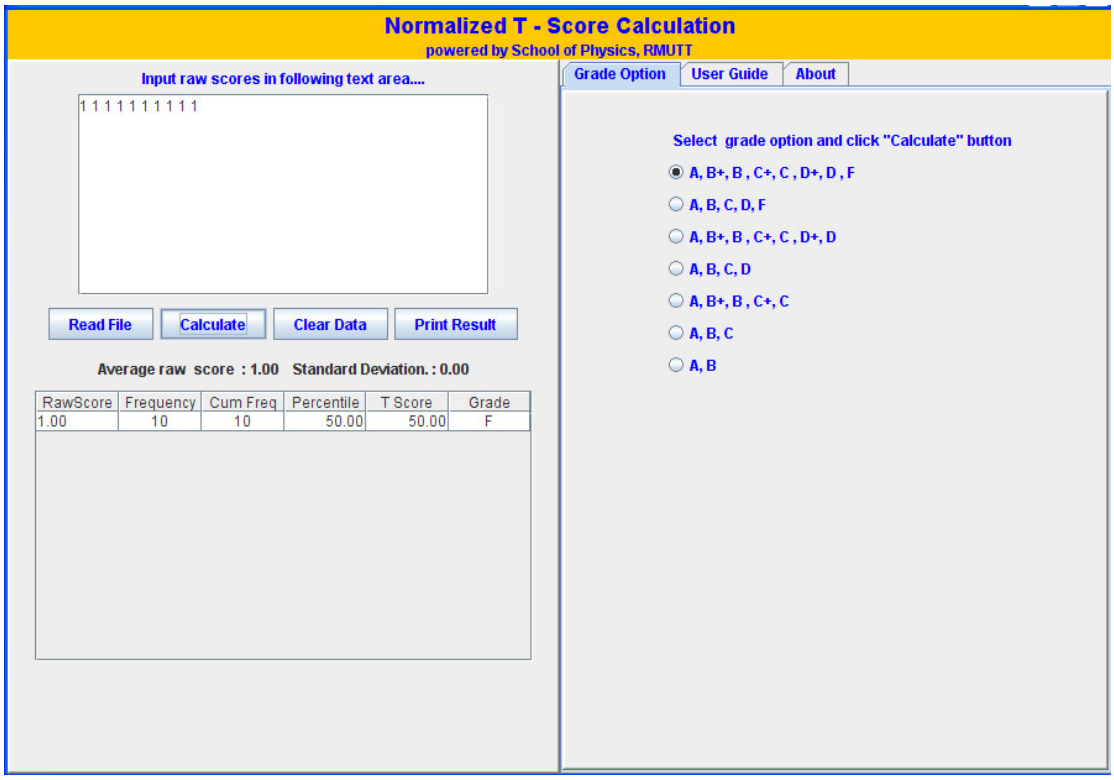
1.3 ให้คะแนน 5,000,000,000



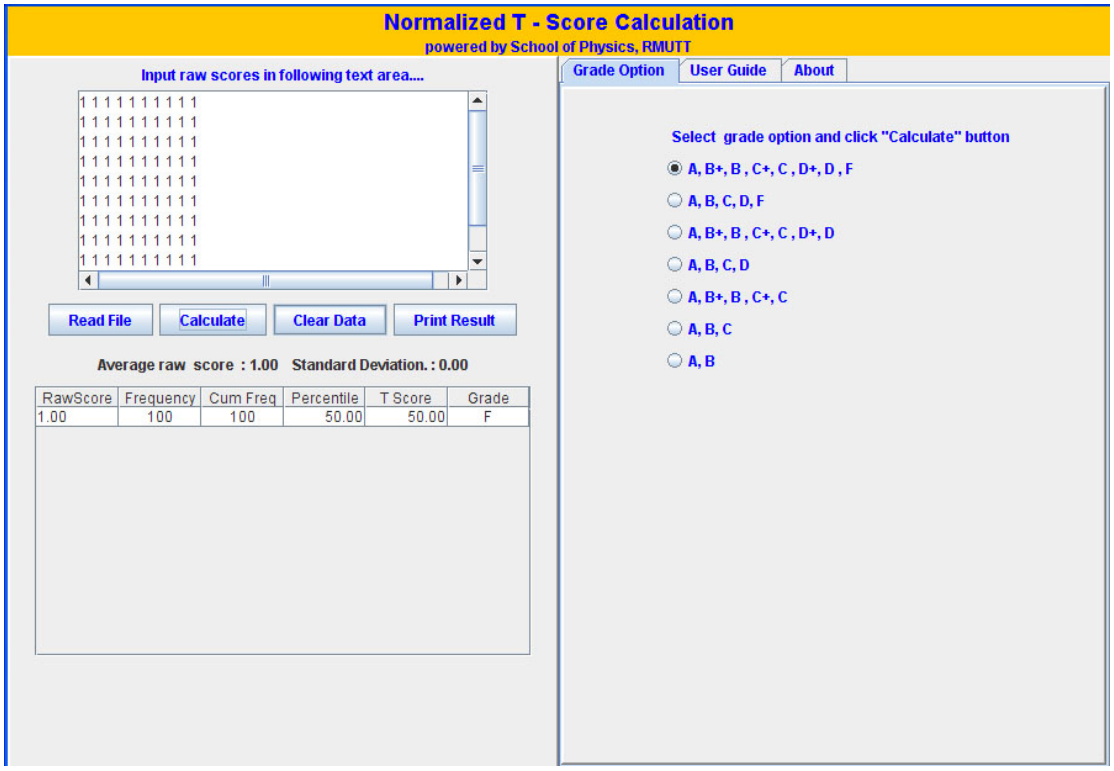
สรุปผลการทดสอบ โปรแกรม Normalized Tscore เมื่อให้คะแนนจำนวน 1 จะได้คะแนน Tscore = 50 โปรแกรมจะตัดเกรดต่ำสุด เช่นถ้าเลือกแบ่งเป็น 4 เกรด A B C และ D ก็จะได้เกรด D

2. การทดสอบโดยให้คะแนนเดียวกันจำนวนหลายๆ ตัว

2.1 ให้คะแนน 1 จำนวน 10 ตัว



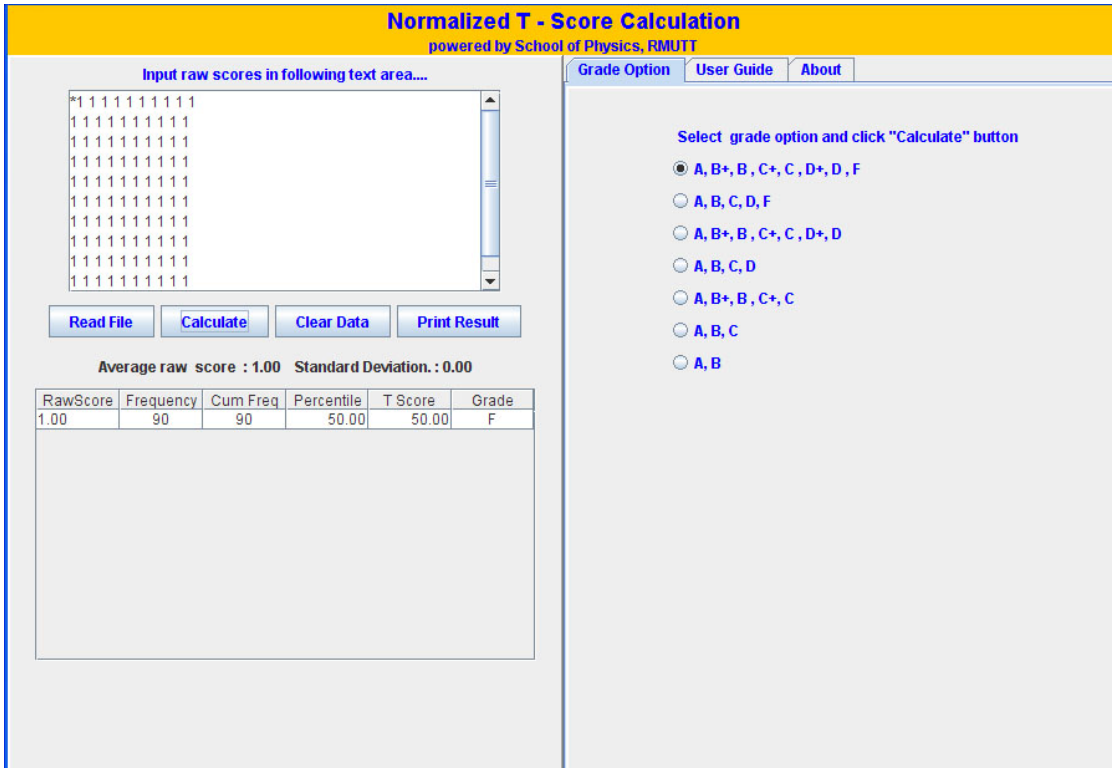
2.2 ให้คะแนนเป็น 1 จำนวน 100 ตัว



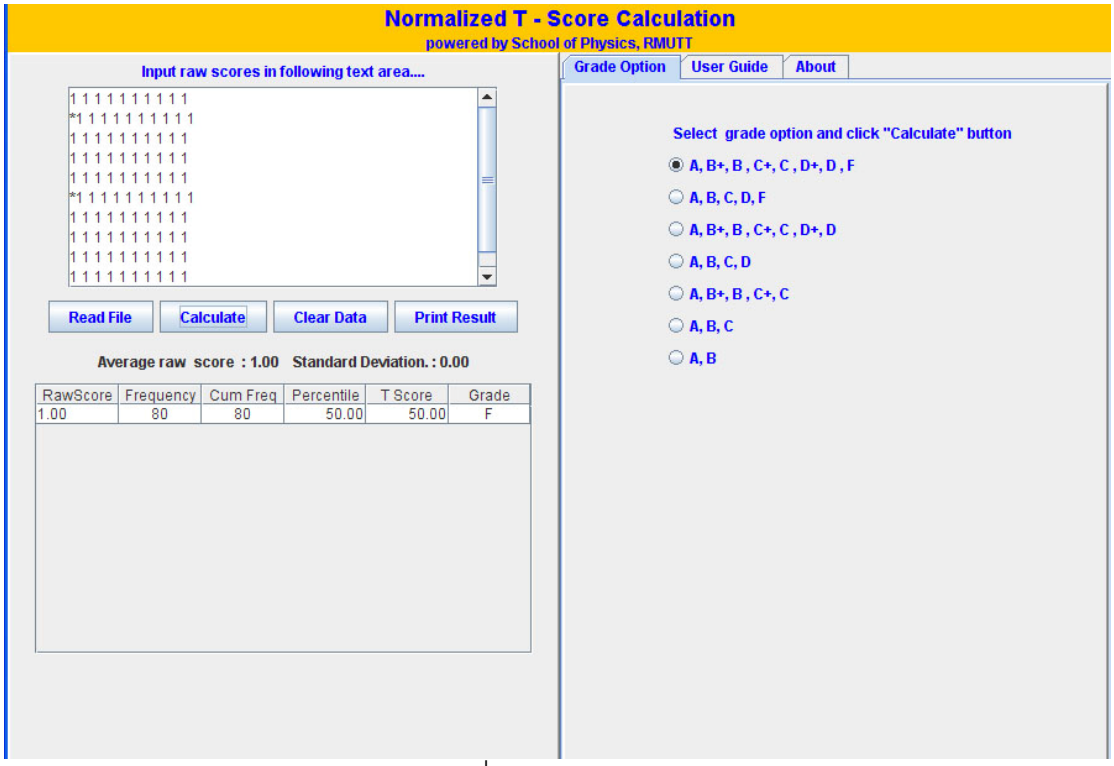
สรุป ถ้าให้คะแนนใดก็แล้วแต่ซ้ำกันจะได้คะแนน Tsore = 50 และได้เกรดต่ำที่สุดในช่วงที่เลือก

3. ทดสอบการใส่เครื่องหมาย * (comment) ที่บรรทัดต่างๆ

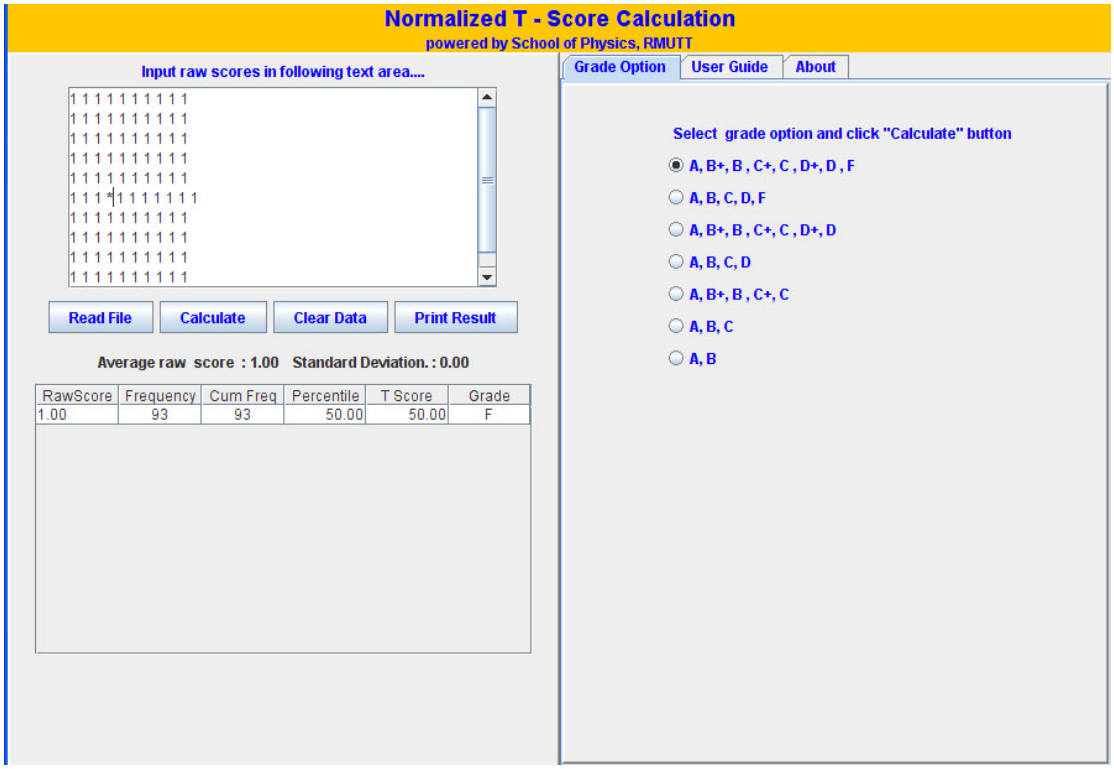
3.1 ใส่ที่บรรทัดแรกของข้อมูล



3.2 ใส่ไว้ที่บรรทัดที่ 2 และ 6

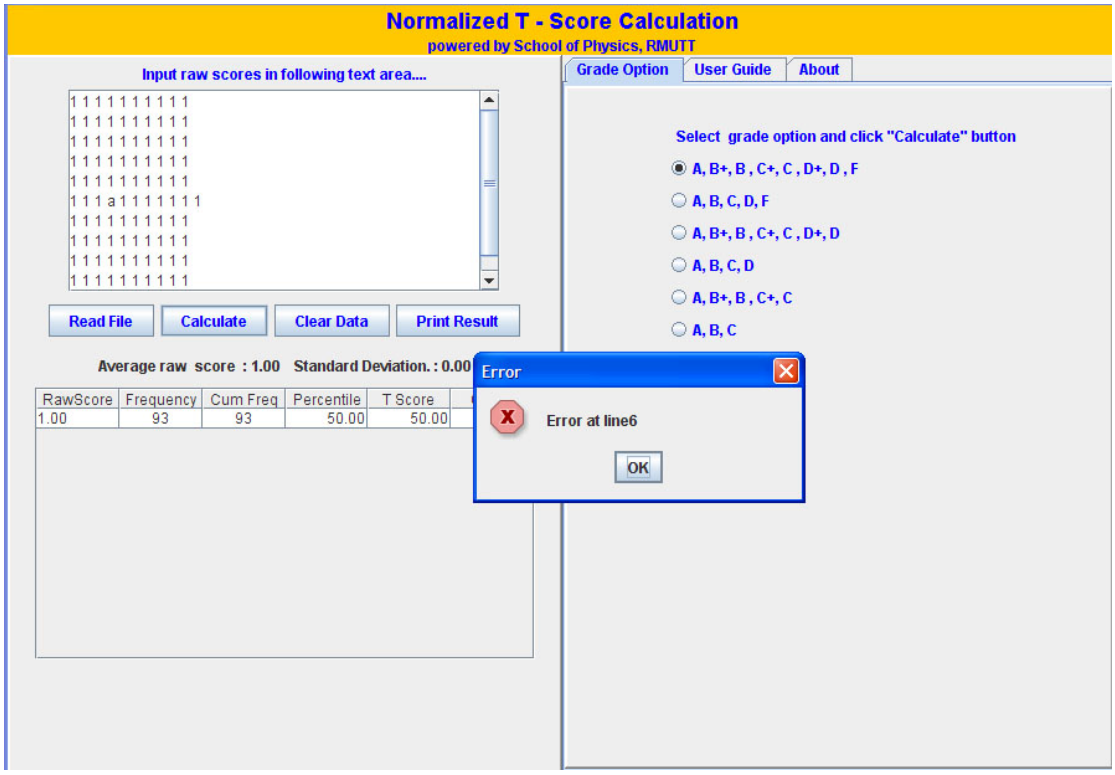


3.3 ใส่ไว้ระหว่างข้อมูลในบรรทัดที่ 5



สรุป โปรแกรม จะไม่นำคะแนนที่อยู่หลังเครื่องหมาย * ในบรรทัดนั้นมาคำนวณ

4. ทดสอบการใส่ตัวอักษรระหว่างบรรทัด



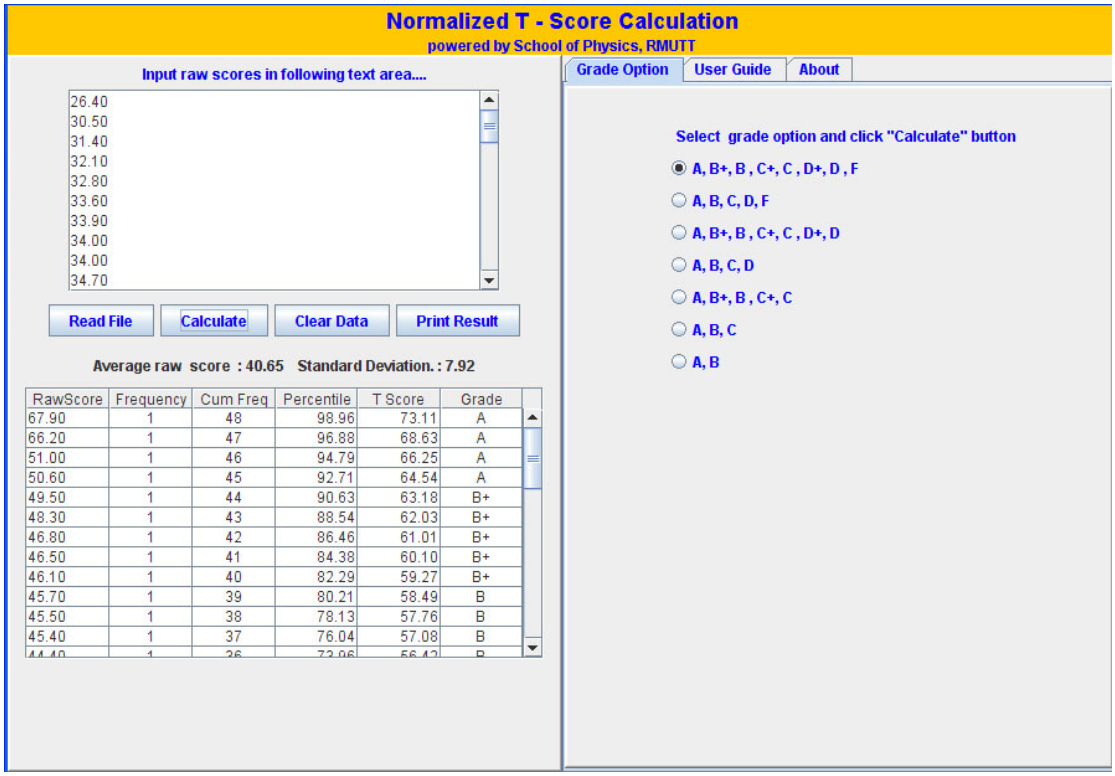
5. ทดสอบการตัดเกรดด้วยโปรแกรม Normalized Tscore version เก่าที่ใช้ในการตัดเกรดในวิชาฟิสิกส์ กับ version ใหม่ ด้วยคะแนนของนักศึกษาวิชาฟิสิกส์พื้นฐาน 1 ภาคเรียนที่ 3 ปีการศึกษา 2550 ผลที่ได้คือ

จากจำนวนนักศึกษา 48 คน

Version เก่า และใหม่ได้ค่าเฉลี่ยของคะแนน 40.65

แต่ได้ค่า Standard deviation ไม่เท่ากันคือ version เก่าได้ 7.83 version ใหม่ได้ 7.92

ผลจากการคำนวณด้วยโปรแกรม Excel ได้ค่า Standard deviation = 7.92



Raw Score	Normalized Tsore					Tscore version เก่า	
	Frequency	Cum Freq	Percentile	Tscore	Grade	Tsore	Grade
67.9	1	48	98.96	73.11	A	70.3	A
66.2	1	47	96.88	68.63	A	66.2	A
51	1	46	94.79	66.25	A	64.9	A
50.6	1	45	92.71	64.54	A	63.6	A
49.5	1	44	90.63	63.18	B+	62.5	A
48.3	1	43	88.54	62.03	B+	61.5	B+
46.8	1	42	86.46	61.01	B+	60.6	B+
46.5	1	41	84.38	60.1	B+	60	B+
46.1	1	40	82.29	59.27	B+	59.5	B+
45.7	1	39	80.21	58.49	B	58.7	B+
45.5	1	38	78.13	57.76	B	57.9	B
45.4	1	37	76.04	57.08	B	57.2	B
44.4	1	36	73.96	56.42	B	56.5	B

44.2	1	35	71.88	55.79	B	55.8	B
44.1	2	34	68.75	54.89	B	54.9	B
43.8	1	32	65.63	54.02	C+	53.9	C+
41.9	1	31	63.54	53.46	C+	53.4	C+
41	1	30	61.46	52.91	C+	52.8	C+
40.6	1	29	59.38	52.37	C+	52.2	C+
39.5	2	28	56.25	51.57	C+	51.4	C+
39.3	1	26	53.13	50.78	C+	50.6	C+
39.2	1	25	51.04	50.26	C+	50.1	C+
38.6	2	24	47.92	49.48	C	49.7	C
38.5	1	22	44.79	48.69	C	48.9	C
38.2	1	21	42.71	48.16	C	48.3	C
37.5	1	20	40.63	47.63	C	47.8	C
37.4	1	19	38.54	47.09	C	47.2	C
37.1	1	18	36.46	46.54	C	46.6	C
36.8	2	17	33.33	45.69	C	45.8	D+
36.3	1	15	30.21	44.82	D+	44.8	D+
35.5	1	14	28.13	44.21	D+	44.2	D+
35.4	1	13	26.04	43.58	D+	43.5	D+
35	2	12	22.92	42.58	D+	42.5	D+
34.7	1	10	19.79	41.51	D+	41.3	D
34	2	9	16.67	40.33	D	40.1	D
33.9	1	7	13.54	38.99	D	39.4	D
33.6	1	6	11.46	37.97	D	38.5	D
32.8	1	5	9.38	36.82	D	37.5	F
32.1	1	4	7.29	35.46	F	36.4	F
31.4	1	3	5.21	33.75	F	35.1	F
30.5	1	2	3.13	31.37	F	33.8	F
26.4	1	1	1.04	26.89	F	29.7	F

6. การตัดเกรดวิชาฟิสิกส์ 2 เทอม 2 ปีการศึกษา 2550 ด้วยโปรแกรม Normalized Tscore จำนวนนักศึกษา 906 คน ค่าเฉลี่ยคะแนน 43.58 ค่า Standard Deviation = 9.64

raw score	Frequency	Cum Freq	Percentile	Tscore	Grade
79.64	1	906	99.94	82.63	A
76.36	1	905	99.83	79.37	A
73.82	1	904	99.72	77.75	A
73.55	1	903	99.61	76.64	A
71.18	1	902	99.5	75.78	A
70.91	1	901	99.39	75.08	A
70.36	1	900	99.28	74.48	A
69.82	2	899	99.12	73.73	A
68.91	1	897	98.95	73.09	A
68.45	1	896	98.84	72.7	A
67.82	1	895	98.73	72.35	A
67.64	3	894	98.51	71.73	A
67.27	1	891	98.29	71.18	A
67.18	1	890	98.18	70.92	A
66.91	2	889	98.01	70.57	A
66.27	1	887	97.85	70.23	A
65.45	1	886	97.74	70.02	A
65.09	1	885	97.63	69.82	A
64.73	2	884	97.46	69.53	B+
64.55	1	882	97.3	69.26	B+
64.36	3	881	97.08	68.92	B+
64.18	2	878	96.8	68.52	B+
64	1	876	96.63	68.29	B+
63.64	1	875	96.52	68.15	B+
62.91	3	874	96.3	67.87	B+

62.55	3	871	95.97	67.47	B+
62.36	1	868	95.75	67.22	B+
62.18	3	867	95.53	66.99	B+
61.45	5	864	95.09	66.53	B+
61.18	1	859	94.76	66.22	B+
61.09	2	858	94.59	66.07	B+
61	1	856	94.43	65.92	B+
60.91	2	855	94.26	65.77	B+
60.73	1	853	94.09	65.63	B+
60.36	2	852	93.93	65.49	B+
60	1	850	93.76	65.35	B+
59.82	1	849	93.65	65.26	B+
59.68	1	848	93.54	65.18	B+
59.64	1	847	93.43	65.09	B+
59.5	1	846	93.32	65	B+
59.45	1	845	93.21	64.92	B+
59.27	2	844	93.05	64.79	B+
59.09	2	842	92.83	64.63	B+
58.91	2	840	92.6	64.47	B+
58.36	1	838	92.44	64.35	B+
58.09	1	837	92.33	64.28	B+
58	1	836	92.22	64.2	B+
57.82	1	835	92.11	64.12	B+
57.68	1	834	92	64.05	B+
57.64	2	833	91.83	63.94	B+
57.45	2	831	91.61	63.79	B+
57.36	1	829	91.45	63.69	B+
57.27	2	828	91.28	63.58	B+
57.09	2	826	91.06	63.44	B+
56.91	1	824	90.89	63.34	B+

56.64	1	823	90.78	63.28	B+
56.55	2	822	90.62	63.18	B+
56.45	1	820	90.45	63.08	B+
56.41	1	819	90.34	63.01	B
56.36	2	818	90.18	62.92	B
56.27	1	816	90.01	62.82	B
56.18	2	815	89.85	62.73	B
56	3	813	89.57	62.57	B
55.82	1	810	89.35	62.45	B
55.64	1	809	89.24	62.39	B
55.45	5	808	88.91	62.22	B
55.36	2	803	88.52	62.01	B
55.27	6	801	88.08	61.79	B
55.09	1	795	87.69	61.6	B
54.91	3	794	87.47	61.49	B
54.73	3	791	87.14	61.33	B
54.64	1	788	86.92	61.23	B
54.55	4	787	86.64	61.1	B
54.36	5	783	86.15	60.87	B
54.23	1	778	85.82	60.72	B
54.18	2	777	85.65	60.65	B
54	1	775	85.49	60.58	B
53.91	2	774	85.32	60.5	B
53.82	2	772	85.1	60.41	B
53.73	1	770	84.93	60.34	B
53.64	2	769	84.77	60.27	B
53.45	2	767	84.55	60.17	B
53.36	1	765	84.38	60.1	B
53.14	1	764	84.27	60.06	B
53.09	3	763	84.05	59.97	B

52.91	2	760	83.77	59.85	B
52.73	1	758	83.61	59.79	B
52.64	2	757	83.44	59.72	B
52.59	1	755	83.28	59.65	B
52.55	1	754	83.17	59.61	B
52.41	1	753	83.06	59.56	B
52.36	5	752	82.73	59.43	B
52.18	7	747	82.06	59.18	B
52.09	1	740	81.62	59.01	B
52	3	739	81.4	58.93	B
51.82	3	736	81.07	58.81	B
51.68	1	733	80.85	58.72	B
51.64	1	732	80.74	58.68	B
51.45	3	731	80.52	58.6	B
51.36	1	728	80.3	58.52	B
51.27	3	727	80.08	58.44	B
51.14	1	724	79.86	58.37	B
51.09	2	723	79.69	58.31	B
51	2	721	79.47	58.23	B
50.91	2	719	79.25	58.15	B
50.82	1	717	79.08	58.09	B
50.77	1	716	78.97	58.06	B
50.73	4	715	78.7	57.96	B
50.55	4	711	78.26	57.81	B
50.41	1	707	77.98	57.72	B
50.36	3	706	77.76	57.64	B
50.27	1	703	77.54	57.57	B
50.18	4	702	77.26	57.48	B
50.05	1	698	76.99	57.38	B
50	3	697	76.77	57.31	B

49.82	6	694	76.27	57.15	B
49.73	1	688	75.88	57.03	B
49.64	3	687	75.66	56.95	B
49.55	2	684	75.39	56.87	B
49.45	6	682	74.94	56.73	B
49.41	1	676	74.56	56.61	B
49.36	1	675	74.45	56.57	B
49.27	1	674	74.34	56.54	B
49.18	1	673	74.23	56.5	C+
49.09	3	672	74.01	56.44	C+
48.95	1	669	73.79	56.37	C+
48.82	1	668	73.68	56.33	C+
48.73	6	667	73.29	56.22	C+
48.59	1	661	72.9	56.1	C+
48.55	1	660	72.79	56.07	C+
48.36	5	659	72.46	55.97	C+
48.18	1	654	72.13	55.87	C+
48.05	1	653	72.02	55.83	C+
48	1	652	71.91	55.8	C+
47.77	1	651	71.8	55.77	C+
47.73	1	650	71.69	55.74	C+
47.64	2	649	71.52	55.69	C+
47.55	1	647	71.36	55.64	C+
47.45	3	646	71.14	55.57	C+
47.27	6	643	70.64	55.43	C+
47.18	1	637	70.25	55.32	C+
47.14	2	636	70.09	55.27	C+
47.09	4	634	69.76	55.17	C+
46.91	8	630	69.09	54.99	C+
46.82	1	622	68.6	54.85	C+

46.77	1	621	68.49	54.81	C+
46.73	2	620	68.32	54.77	C+
46.64	3	618	68.05	54.69	C+
46.55	1	615	67.83	54.63	C+
46.36	2	614	67.66	54.58	C+
46.18	5	612	67.27	54.48	C+
46.09	2	607	66.89	54.37	C+
46.05	4	605	66.56	54.28	C+
46	1	601	66.28	54.2	C+
45.91	2	600	66.11	54.16	C+
45.82	6	598	65.67	54.04	C+
45.73	2	592	65.23	53.92	C+
45.64	3	590	64.96	53.84	C+
45.55	3	587	64.62	53.75	C+
45.45	6	584	64.13	53.62	C+
45.36	1	578	63.74	53.52	C+
45.27	6	577	63.36	53.41	C+
45.18	3	571	62.86	53.28	C+
45.09	8	568	62.25	53.12	C+
44.91	4	560	61.59	52.95	C+
44.82	5	556	61.09	52.82	C+
44.73	8	551	60.38	52.63	C+
44.55	4	543	59.71	52.46	C+
44.45	2	539	59.38	52.37	C+
44.36	6	537	58.94	52.26	C+
44.27	3	531	58.44	52.13	C+
44.18	5	528	58	52.02	C+
44.09	2	523	57.62	51.92	C+
44	6	521	57.17	51.81	C+
43.86	1	515	56.79	51.71	C+

43.82	5	514	56.46	51.63	C+
43.64	5	509	55.91	51.49	C+
43.55	4	504	55.41	51.36	C+
43.5	1	500	55.13	51.29	C+
43.45	5	499	54.8	51.21	C+
43.36	4	494	54.3	51.08	C+
43.27	7	490	53.7	50.93	C+
43.18	2	483	53.2	50.8	C+
43.09	4	481	52.87	50.72	C+
43	2	477	52.54	50.64	C+
42.91	7	475	52.04	50.51	C+
42.82	2	468	51.55	50.39	C+
42.77	1	466	51.38	50.35	C+
42.73	4	465	51.1	50.28	C+
42.64	3	461	50.72	50.18	C+
42.55	3	458	50.39	50.1	C+
42.45	2	455	50.11	50.03	C+
42.36	4	453	49.78	49.94	C
42.27	3	449	49.39	49.85	C
42.23	1	446	49.17	49.79	C
42.18	5	445	48.84	49.71	C
42.09	1	440	48.51	49.63	C
42	1	439	48.4	49.6	C
41.91	7	438	47.96	49.49	C
41.82	4	431	47.35	49.34	C
41.77	1	427	47.08	49.27	C
41.73	1	426	46.96	49.24	C
41.64	5	425	46.63	49.16	C
41.55	3	420	46.19	49.04	C
41.45	8	417	45.58	48.89	C

41.36	1	409	45.09	48.77	C
41.27	3	408	44.87	48.71	C
41.18	5	405	44.43	48.6	C
41.09	3	400	43.98	48.49	C
41	4	397	43.6	48.39	C
40.95	2	393	43.27	48.3	C
40.91	5	391	42.88	48.21	C
40.82	1	386	42.55	48.12	C
40.73	2	385	42.38	48.08	C
40.64	2	383	42.16	48.02	C
40.55	2	381	41.94	47.97	C
40.36	8	379	41.39	47.82	C
40.18	3	371	40.78	47.67	C
40.09	1	368	40.56	47.61	C
40	3	367	40.34	47.55	C
39.82	2	364	40.07	47.48	C
39.73	8	362	39.51	47.34	C
39.68	1	354	39.02	47.21	C
39.64	6	353	38.63	47.11	C
39.55	4	347	38.08	46.97	C
39.45	7	343	37.47	46.81	C
39.36	1	336	37.03	46.69	C
39.27	3	335	36.81	46.63	C
39.18	3	332	36.48	46.54	C
39.09	4	329	36.09	46.44	C
39	2	325	35.76	46.35	C
38.91	6	323	35.32	46.23	C
38.82	2	317	34.88	46.11	C
38.73	2	315	34.66	46.05	C
38.64	2	313	34.44	45.99	C

38.55	11	311	33.72	45.8	C
38.45	1	300	33.06	45.62	C
38.36	6	299	32.67	45.51	C
38.27	4	293	32.12	45.36	C
38.23	1	289	31.84	45.28	C
38.18	5	288	31.51	45.19	C
38.14	1	283	31.18	45.09	C
38.09	1	282	31.07	45.06	C
38.05	1	281	30.96	45.03	C
38	1	280	30.85	45	C
37.91	8	279	30.35	44.86	C
37.82	3	271	29.75	44.68	C
37.73	3	268	29.42	44.59	C
37.64	7	265	28.86	44.43	C
37.55	4	258	28.26	44.25	C
37.45	7	254	27.65	44.07	C
37.27	4	247	27.04	43.88	C
37.18	3	243	26.66	43.77	C
37.09	6	240	26.16	43.62	C
37	2	234	25.72	43.48	C
36.91	4	232	25.39	43.38	D+
36.86	1	228	25.11	43.29	D+
36.82	3	227	24.89	43.22	D+
36.73	6	224	24.39	43.06	D+
36.68	1	218	24.01	42.94	D+
36.64	5	217	23.68	42.83	D+
36.55	7	212	23.01	42.62	D+
36.45	4	205	22.41	42.41	D+
36.41	2	201	22.08	42.3	D+
36.36	6	199	21.63	42.15	D+

36.27	2	193	21.19	42	D+
36.18	4	191	20.86	41.89	D+
36.09	5	187	20.36	41.71	D+
36	7	182	19.7	41.48	D+
35.91	3	175	19.15	41.28	D+
35.86	1	172	18.93	41.19	D+
35.82	3	171	18.71	41.11	D+
35.73	1	168	18.49	41.03	D+
35.68	1	167	18.38	40.99	D+
35.64	4	166	18.1	40.88	D+
35.59	1	162	17.83	40.78	D+
35.55	2	161	17.66	40.72	D+
35.45	4	159	17.33	40.59	D+
35.36	2	155	17	40.46	D+
35.32	1	153	16.83	40.39	D+
35.27	6	152	16.45	40.24	D+
35.18	1	146	16.06	40.08	D+
35.14	1	145	15.95	40.03	D+
35.09	5	144	15.62	39.9	D+
35	3	139	15.18	39.71	D+
34.91	3	136	14.85	39.57	D+
34.82	2	133	14.57	39.45	D+
34.73	3	131	14.29	39.33	D+
34.64	5	128	13.85	39.13	D+
34.55	6	123	13.25	38.85	D+
34.41	1	117	12.86	38.67	D+
34.36	3	116	12.64	38.56	D+
34.27	3	113	12.31	38.4	D+
34.18	7	110	11.75	38.13	D+
34	5	103	11.09	37.78	D+

33.91	3	98	10.65	37.55	D+
33.82	4	95	10.26	37.33	D+
33.73	3	91	9.88	37.11	D+
33.68	1	88	9.66	36.99	D+
33.64	2	87	9.49	36.89	D
33.55	1	85	9.33	36.79	D
33.45	3	84	9.11	36.66	D
33.36	1	81	8.89	36.52	D
33.27	2	80	8.72	36.42	D
33.14	3	78	8.44	36.24	D
33.09	4	75	8.06	35.99	D
33	1	71	7.78	35.8	D
32.82	2	70	7.62	35.69	D
32.73	3	68	7.34	35.49	D
32.55	3	65	7.01	35.25	D
32.45	2	62	6.73	35.04	D
32.36	1	60	6.57	34.91	D
32.27	2	59	6.4	34.78	D
32.18	5	57	6.02	34.47	D
32.09	2	52	5.63	34.13	D
32	1	50	5.46	33.99	D
31.73	2	49	5.3	33.83	D
31.64	2	47	5.08	33.63	D
31.55	1	45	4.91	33.47	D
31.45	2	44	4.75	33.3	D
31.36	1	42	4.58	33.13	D
31.14	1	41	4.47	33.01	D
31.09	2	40	4.3	32.84	D
31	3	38	4.03	32.53	D
30.91	1	35	3.81	32.27	D

30.73	1	34	3.7	32.13	D
30.64	3	33	3.48	31.85	D
30.27	2	30	3.2	31.48	D
29.91	1	28	3.04	31.24	D
29.73	1	27	2.92	31.08	D
29.68	1	26	2.81	30.91	D
29.64	2	25	2.65	30.65	D
29.55	1	23	2.48	30.37	F
29.45	2	22	2.32	30.08	F
29.36	3	20	2.04	29.55	F
29.18	1	17	1.82	29.08	F
28.91	1	16	1.71	28.82	F
28.64	1	15	1.6	28.56	F
28.36	1	14	1.49	28.27	F
27.82	1	13	1.38	27.97	F
27.73	1	12	1.27	27.65	F
27.45	1	11	1.16	27.3	F
23.09	1	10	1.05	26.91	F
19	1	9	0.94	26.5	F
16.73	1	8	0.83	26.04	F
8.73	1	7	0.72	25.52	F
7.27	1	6	0.61	24.92	F
6.55	1	5	0.5	24.22	F
5.82	2	4	0.33	22.85	F
5.45	1	2	0.17	20.63	F
5.09	1	1	0.06	17.37	F

INDEX

A

AboutPanel, class 50, 51, 62
AboutPanel.java 62
Absolute zero 6
Apache 23, 121, 127
Applet 2, 3, 4, 64
archives 66
Array 28
assessGrade, method 41

B

BorderLayout 50
ButtonPanel 51
byte code 3, 23, 27

C

certificate 118
class, file 23
codebase 65
Common, folder 24
computeTScore, method 41
configure, server 77, 121
Correlation 13
CountIF 102
criterion reference 8, 9, 10

D

DEFAULT_TOLERANCE 80
Derived score 15

E

Eclipse 23
Editor 23
Evaluation 6
Excel 25, 72, 73, 74, 78, 87, 94, 99, 103, 115

F

findAreaUnderNormalCurve 38
findCumulativeFrequency 38
findFrequency 38
findMean 38
findSD 38
findZAtKnownArea 38, 41, 79, 92
finger print 119
Formative evaluation 8
Frequency distribution 10

G

GradeOptionPanel, class 46, 50, 51
GradeOptionPanel.java 47
GridLayout 50
grouped data 11

H

HelpPanel, class 50, 51
HelpPanel.java 59

I

IIS 122, 128
Integration 24,25
InverseZArea.java 93

J

jar, command 65, 67
jar, file 27, 65, 66, 77, 117, 125
jarsigner 67, 119
Java 27, 75
Java Network Launchig Protocol 76
Java Runtime Environment 3
Java SDK 23
Java Virtual Machine 3
Java Web Start 77, 117
Jcreator light 23
jnlp, file 123,124
JRE 3, 76, 130
JTextArea 28
JVM 3

K

keystore 118
keytool 67, 117

L

LAN 84
Linux 3

M

MacOS 3
manifest, file 66, 117, 119
MathTools 23
Mean 11
Mean deviation 12
Measure of control tendency 11
Measure of variability 12
Measurement 5
Median 12
Mode 12, 38
Mode 38

N

Norm reference 8, 9, 10
Normalized T score 1, 2, 23
NormSDist 41, 87
NormSInv 41, 72, 94, 108
Notepad 23
NULL 78

O

Object Oriented, Language 3
Object Oriented, Programming 27
Open source 23

P

Percentile 3, 106, 133
Pre evaluation 7
Printable, class 58
PrinterJob, class 58

Q

Quick sort 81
quickSort, method 38, 81,82

R

range 11, 12
range, formula 45
Raw score 15

S

ScoreTextArea 51
ScoreTextArea, class 28, 31
ScoreTextArea.java 29
SimpleStat, class 31, 79
SimpleStat, Exception 38
SimpleStat, folder 25
SimpleStat.java 31
Solaris 3
Standard deviation 12
Standard normal distribution 17
Standard score 16
StatPanel 51
Summative evaluation 8

T

TabbedPane 49, 50
TablePanel 51
tally 11
Text Area 133
Text area, box 27
Text area, ปัญหา 77
Text Editor 132
Tolerance 78,81
Tscore, class 41
Tscore.java 42
TScoreApp.java 66
TScoreApplet.java 64

TScoreCal.jnlp 68, 123
TScoreConstants, class 28
TScoreConstants.java 29
TScorePanel, class 31, 41, 50, 52, 58

U

ungrouped data 11
User guide 49, 59, 134

V

Variance 12
VBA 74
Visual Basic Application 74
Visual Basic Editor 110, 111

W

Web master 127
Web server 23
Windows 3

Z

Z, คะแนน 16, 25
Z, ค่า 92, 93
Z, ตาราง 95, 96, 97, 98

ก

กรอบป้อนข้อความ 27
กรอบป้อนข้อความ, รูป 28
กำหนดระดับคะแนน, การ 18

ค

ความถี่ 69, 102, 115, 133
ความถี่สะสม 69, 104, 105, 115, 133
ความแปรปรวน 12
คอมไพเลอร์ 3
คะแนน Z 16
คะแนน, ความหมาย 14
คะแนนดิบ, ลักษณะ 15
คะแนนที 3, 16
คะแนนทีปกติ 108, 109, 110, 133
คะแนนทีปกติ 1, 2, 17, 20, 21, 23, 25, 45, 46, 69, 71
คะแนนทีปกติ, ตาราง 70
คะแนนทีปกติ, ประโยชน์ 18
คะแนนแปลงรูป 15
คะแนนมาตรฐาน 16
ค่าเฉลี่ย 11
คู่มือการใช้งาน 127
เครื่องมือวัด 5

โค้งปกติ 17, 20
โค้งปกติ, พื้นที่ 23, 25
โค้งปกติ, รูป 18

จ

จาวา 2, 3, 23, 75
แจกแจงความถี่, การ 10
แจกแจงความถี่แบบจัดหมวดหมู่ 11
แจกแจงความถี่แบบไม่จัดหมวดหมู่ 11

ซ

ซันไมโคร ซิสเท็ม, บริษัท 3

ฐ

ฐานนิยม 12

ต

ตัดเกรด, การ 18, 21, 22
ตัดเกรด, องค์ประกอบ 19
ตัดสินคุณค่า 6

ป

ประเมินผล, การวัด 6, 7
ประเมินผลก่อนเรียน 7
ประเมินผลระหว่างเรียน 8
ประเมินผลสรุป 8
ปริพันธ์, การหา 25
เปอร์เซ็นต์ไทล์ 106
เปอร์เซ็นต์ไทล์ 3
เปอร์เซ็นต์ไทล์ 15

พ

พิสัย 11, 12
พิสัย, สูตร 45
พื้นที่ใต้โค้ง 87
พื้นที่ใต้โค้ง, ตาราง 40, 88, 89, 90,91

ภ

ภาษาเชิงวัตถุ 3

ม

มัธยฐาน 12

ว

วัดการกระจาย, การ 12
วัดแนวโน้มเข้าสู่ส่วนกลาง 11
วัดผล, การ 5, 6
วินโดว์ 23

ศ

ศูนย์แท้ 6
ศูนย์สัมบูรณ์ 6

ส

ส่วนเบี่ยงเบนเฉลี่ย 12
ส่วนเบี่ยงเบนมาตรฐาน 12
สหสัมพันธ์ 13, 14
เส้นโค้งปกติ, สมการ 87

อ

อะเรย์ 28, 38
อิงกลุ่ม, การตัดเกรด 20
อิงกลุ่ม, ประเมินผล 8, 9, 10
อิงเกณฑ์, การตัดเกรด 19
อิงเกณฑ์, ประเมินผล 8, 9, 10
แอพเพล็ต 2, 3, 4, 26, 64