

แผนบริหารการสอนประจำบทที่ 1

หัวข้อเนื้อหา

- ประวัติการเขียนโปรแกรมคอมพิวเตอร์
- หลักการและแนวคิดการเขียนโปรแกรมคอมพิวเตอร์
- ขั้นตอนการเขียนโปรแกรมคอมพิวเตอร์
- ผังงานการเขียนโปรแกรมคอมพิวเตอร์

วัตถุประสงค์เชิงพฤติกรรม

หลังจากศึกษาบทที่ 1 แล้วนักเรียนมีความสามารถดังต่อไปนี้

1. ทราบประวัติความเป็นมาการเขียนโปรแกรมคอมพิวเตอร์
2. อธิบายหลักการและแนวคิดการเขียนโปรแกรมคอมพิวเตอร์ได้อย่างถูกต้อง
3. อธิบายขั้นตอนและผังงานการเขียนโปรแกรมคอมพิวเตอร์ได้อย่างถูกต้อง

วิธีการสอน และกิจกรรม

1. บรรยาย
2. ฝึกปฏิบัติ
3. มอบหมายงานให้ไปปฏิบัติ

สื่อการเรียนการสอน

1. ไฟล์นำเสนอบทที่ 1
2. เอกสารประกอบการสอนบทที่ 1

การวัดและประเมินผล

การวัดผล

- | | |
|------------------------|-----|
| 1. ตอบคำถามในชั้นเรียน | 30% |
| 2. แบบฝึกหัด | 70% |

การประเมินผล

1. การสังเกตพฤติกรรมการเรียนรู้
2. การประเมินผลจากความสนใจและการอภิปราย
3. การทำแบบฝึกหัด

บทที่ 1

แนวคิดการเขียนโปรแกรม

การเขียนโปรแกรมคอมพิวเตอร์เป็นพื้นฐานในการเขียนโปรแกรมและการพัฒนาเกมคอมพิวเตอร์ การเข้าใจหลักการเขียนโปรแกรมคอมพิวเตอร์ที่มีความจำเป็นอย่างมาก เพื่อให้ผลลัพธ์จากการพัฒนาโปรแกรมคอมพิวเตอร์ที่ต้องการมีความถูกต้องและรวมถึงการตรวจสอบโปรแกรมคอมพิวเตอร์ที่มีการพัฒนามาก่อนแล้ว เนื้อหาในบทนี้จะเป็นการทบทวนความรู้พื้นฐานในการเขียนโปรแกรมคอมพิวเตอร์

โดยเนื้อหาในบทนี้ประกอบด้วย

1. ประวัติการเขียนโปรแกรมคอมพิวเตอร์
2. หลักการและแนวคิดการเขียนโปรแกรมคอมพิวเตอร์
3. ขั้นตอนการเขียนโปรแกรมคอมพิวเตอร์
4. ผังงานการเขียนโปรแกรมคอมพิวเตอร์

1.1 ประวัติการเขียนโปรแกรมคอมพิวเตอร์

การเขียนโปรแกรมคอมพิวเตอร์ เริ่มมีการพัฒนามาตั้งแต่ปี ค.ศ. 1940 โปรแกรมภาษาแอสเซมบลี เป็นภาษาแรกที่ถูกพัฒนาขึ้น

ในปี ค.ศ.1949 John Mauchly's Short Code นำเสนอโปรแกรมภาษาชั้นสูงสำหรับเครื่องคอมพิวเตอร์

ปี ค.ศ. 1954 มีการพัฒนาโปรแกรมภาษาฟอร์แทรน โดยบริษัท IBM ภาษาฟอร์แทรน (Fortran programming language หรือ FORTRAN) เป็นภาษาที่เก่าแก่ที่สุดของวงการคอมพิวเตอร์ ถูกพัฒนาขึ้นในปี ค.ศ. 1950 นิยมนำไปใช้ในการคำนวณทางคณิตศาสตร์และวิทยาศาสตร์

ปี ค.ศ. 1959 มีการพัฒนาโปรแกรมภาษาโคบอล (Cobol) เป็นภาษาโปรแกรมระดับสูง ภาษาหนึ่งที่อยู่มาอย่างยาวนาน COBOL ย่อมาจาก Common Business Oriented Language เป็นภาษาที่นิยมนำไปใช้ทางธุรกิจ

ปี ค.ศ. 1959 มีการพัฒนาโปรแกรมภาษาอาร์พีจี (RPG) ย่อมาจากคำว่า Report Program Generator เป็นภาษาโปรแกรมเพื่อใช้ในการพัฒนาแอปพลิเคชันทางธุรกิจ โดยมีวัตถุประสงค์เริ่มแรกเป็นโปรแกรมเพื่อรายงาน โดยภาษาอาร์พีจี ถูกใช้กับระบบคอมพิวเตอร์เซิร์ฟเวอร์ของไอบีเอ็ม

ปี ค.ศ. 1964 มีการพัฒนาโปรแกรมภาษาเบสิก (BASIC) ย่อมาจาก Beginner's All-purpose Symbolic Instruction Code เป็นภาษาโปรแกรมที่ออกแบบมาให้ใช้งานได้ง่าย

ปี ค.ศ.1972มีการพัฒนาโปรแกรมภาษาซี C โดย Dennis Ritchie แห่ง Bell Labs โดยภาษาซีนั้นพัฒนามาจาก ภาษา B และจากภาษา BCPL ซึ่งในช่วงแรกนั้นภาษาซีถูกออกแบบให้ใช้เป็นภาษาการเขียนโปรแกรมในระบบ UNIX และเริ่มมีคนสนใจมากขึ้นในปี ค.ศ.1978 เมื่อ Brian Kernighan ร่วมกับ Dennis Ritchie พัฒนามาตรฐานของภาษาซีขึ้นมา คือ K&R (Kernighan &Ritchie) และทั้งสองยังได้แต่งหนังสือชื่อว่า “The C Programming Language” โดยภาษาซีนั้นสามารถจะปรับใช้กับเครื่องคอมพิวเตอร์รูปแบบต่างๆได้ ต่อมาในช่วง ปี ค.ศ.1988 Ritchie และ Kernighan ได้ร่วมกับ ANSI (American National Standards Institute) สร้างเป็นมาตรฐานของภาษาซีขึ้นมาใหม่มีชื่อว่า “ANSI C”

ปี ค.ศ.1972มีการพัฒนาโปรแกรมภาษา Prologมาจาก PROgrammationen LOGique (logic programming) สร้างขึ้นโดย Alain Colmerauerราว ค.ศ. 1972 ภาษาโปรล็อกเกิดจากความพยายามที่จะสร้างภาษาที่อาศัยวิธีการทางตรรกศาสตร์แทนที่จะกำหนดคำสั่งอย่างละเอียดให้กับคอมพิวเตอร์

ปี ค.ศ.1978 มีการพัฒนาโปรแกรมภาษา SQL (a query language, later extended) เป็นภาษาสอบถามที่นิยมมากที่สุดของการจัดการฐานข้อมูล สำหรับสร้าง แก้ไขและเรียกใช้ฐานข้อมูล โดยใช้มาตรฐานของแอนซี (ANSI) และ ไอเอสโอ (ISO) ปัจจุบันการใช้งานใช้ในหลายจุดประสงค์มากกว่าใช้สำหรับจัดการโปรแกรมเชิงวัตถุที่เป็นจุดประสงค์แรกของการสร้างภาษาสอบถามเชิงโครงสร้าง

ปี ค.ศ.1991มีการพัฒนาโปรแกรมภาษา Python เป็นภาษาโปรแกรมระดับสูง เพื่อใช้งานทั่วไป[3][4] แบบอินเทอร์พรีเตอร์ ที่สร้างโดย กีโด ฟาน รอสซัม (Guido van Rossum)เป็นภาษาสคริปต์ ทำให้ใช้เวลาในการเขียนและคอมไพล์ไม่มาก ทำให้เหมาะกับงานด้านการดูแลระบบ (System administration) เป็นอย่างยิ่ง ได้มีการสนับสนุนภาษาไพทอนโดยเป็นส่วนหนึ่งของระบบปฏิบัติการยูนิกซ์, ลินุกซ์ และสามารถติดตั้งให้ทำงานเป็นภาษาสคริปต์ของวินโดวส์

ปี ค.ศ.1991มีการพัฒนาโปรแกรมภาษา Visual Basicเป็นภาษาโปรแกรมแบบ GUI สร้างโดยบริษัทไมโครซอฟท์ ภาษานี้เป็นหนึ่งในภาษาโปรแกรมยอดนิยมสำหรับโปรแกรมที่ใช้ในด้านธุรกิจ

ปี ค.ศ.1993มีการพัฒนาโปรแกรมภาษา Ruby เป็นภาษาโปรแกรมเชิงวัตถุ ที่ได้รับอิทธิพลของโครงสร้างภาษามาจากภาษาเพิร์ลกับภาษาเอดา มีความสามารถในเชิงวัตถุแบบเดียวกับภาษาสมอลทอล์ค และมีความสามารถหลายอย่างจากภาษาไพทอน, ภาษาลิสป์, ภาษา Dylan และภาษา CLU ตัวแปลภาษารูบี้ตัวหลักเป็นซอฟต์แวร์เสรี และเป็นตัวแปลแบบอินเทอร์พรีเตอร์

ปี ค.ศ.1995มีการพัฒนาโปรแกรมภาษา Javaเป็นภาษาโปรแกรมเชิงวัตถุพัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่ ซัน ไมโครซิสเต็มส์ ภาษาจาวาถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งของโครงการกรีน (the Green Project) ภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส (C++) โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C)

ปี ค.ศ.1995มีการพัฒนาโปรแกรมภาษา Delphi (Object Pascal) เป็นซอฟต์แวร์ตัวแปลภาษาปาสกาล (Pascal) ตัวหนึ่ง ที่พัฒนาโดยบริษัท Borland ซึ่งรูปแบบของภาษานั้นได้ถูก

ออกแบบระบบภาษาใหม่ เพื่อสนับสนุนการเขียนโปรแกรมแบบเชิงวัตถุ คือ ให้ง่ายต่อการพัฒนาต่อ และสามารถนำไปใช้ได้ (reusable)

ปี ค.ศ.1995มีการพัฒนาโปรแกรมภาษา JavaScript เป็นภาษาสคริปต์ ที่มีลักษณะการเขียนแบบโปรโตไทป์ (Prototyped-based Programming) ส่วนมากใช้ในหน้าเว็บเพื่อประมวลผลข้อมูลที่ฝั่งของผู้ใช้งาน แต่ก็ยังมีใช้เพื่อเพิ่มเติมความสามารถในการเขียนสคริปต์โดยฝังอยู่ในโปรแกรมอื่น ๆ

ปี ค.ศ.1995มีการพัฒนาโปรแกรมภาษา PHPถูกสร้างขึ้นโดย RasmusLerdorfต่อมามีผู้ให้ความสนใจเป็นจำนวนมาก จึงได้ออกเป็นแพ็คเกจ "Personal Home Page" ซึ่งเป็นที่มาของ PHP โดยภาษา PHP เป็นแบบ Server Side Script และเป็น Open Source ที่ผู้ใช้ทั่วไปสามารถดาวน์โหลด Source Code และโปรแกรมใช้ได้ฟรี

ปี ค.ศ. 2000มีการพัฒนาโปรแกรมภาษา ActionScript เป็นภาษาสคริปต์ที่มีพื้นฐานมาจากเอ็กมาสคริปต์ (ECMAScript) ซึ่งเป็นตระกูลเดียวกับจาวาสคริปต์ ใช้สำหรับการพัฒนาเว็บไซต์หรือซอฟต์แวร์ที่ทำงานด้วยแฟลช เพื่อใช้คำนวณและควบคุมการทำงานในตัวแฟลช และให้สามารถโต้ตอบกับผู้ใช้ได้(Interactive)

ปี ค.ศ. 2001มีการพัฒนาโปรแกรมภาษา C# เป็นภาษาโปรแกรมแบบหลายโมเดล ที่ใช้ระบบชนิดข้อมูลแบบรัดกุม (strong typing) และสนับสนุนการเขียนโปรแกรมเชิงคำสั่ง การเขียนโปรแกรมเชิงประกาศ การเขียนโปรแกรมเชิงฟังก์ชัน การเขียนโปรแกรมเชิงกระบวนการ การเขียนโปรแกรมเชิงวัตถุ (แบบคลาส) และการเขียนโปรแกรมเชิงส่วนประกอบ พัฒนาเริ่มแรกโดยบริษัทไมโครซอฟท์เพื่อทำงานบนดอตเน็ตเฟรมเวิร์ก โดยมีแอนเดอร์เฮลส์เบิร์ก (Anders Hejlsberg) เป็นหัวหน้าโครงการ และมีรากฐานมาจากภาษาซีพลัสพลัสและภาษาอื่นๆ (โดยเฉพาะภาษาเดลไฟและจาวา) โดยมีจุดมุ่งหมายให้เป็นภาษาสมัยใหม่ที่ไม่ซับซ้อน ใช้งานได้ทั่วไป (general-purpose) และเป็นเชิงวัตถุเป็นหลัก

ปี ค.ศ. 2001มีการพัฒนาโปรแกรมภาษา Visual Basic .NET เป็นภาษาโปรแกรมเชิงวัตถุทำงานบนดอตเน็ตเฟรมเวิร์ก พัฒนามาจากภาษาวิซวลเบสิก

ปี ค.ศ. 2003มีการพัฒนาโปรแกรมภาษา Groovy เป็นภาษาโปรแกรมเชิงวัตถุสำหรับแพลตฟอร์มจาวา ลักษณะเป็นเหมือนภาษาสคริปต์สำหรับแพลตฟอร์มจาวา เนื่องจากมีคุณลักษณะหลายอย่างเหมือนกับภาษาสคริปต์อย่าง ไพทอน (Python) รูบี้ (Ruby) เพิร์ล (Perl) และ สمولทอล์ค (Smalltalk) อาจจะใช้ชื่อเรียกJSR 241แทนภาษากรูวีโปรแกรมที่เขียนด้วยภาษากรูวี จะถูกคอมไพล์ เป็นจาวาไบต์โค้ด ซึ่งสามารถใช้งานได้บนจาวาเวอร์ชวลแมชีน (JVM)

ปี ค.ศ. 2003มีการพัฒนาโปรแกรมภาษา Scala ลักษณะของภาษารองรับการเขียนโปรแกรมทั้งแบบ functional และ imperative ทั้งสองแบบ ตัวภาษา Scala เป็นภาษาที่ทำงานอยู่บน JVM ทำให้สามารถเรียกใช้งานไลบรารีทั้งหมดที่สามารถทำงานบน JVM ได้ทันที

ปี ค.ศ. 2005มีการพัฒนาโปรแกรมภาษา F# เป็นภาษาสำหรับเขียนโปรแกรมคอมพิวเตอร์แบบ strongly typed บนดอตเน็ตเฟรมเวิร์กและ Mono ที่สนับสนุนโมเดลการเขียนโปรแกรมหลาย

แบบรวมทั้ง การเขียนโปรแกรมเชิงฟังก์ชัน การเขียนโปรแกรมเชิงวัตถุ และการเขียนโปรแกรมเชิงคำสั่ง F# เป็นภาษาดอตเน็ต สามารถเรียกใช้เอพีไอของดอตเน็ต และถูกเรียกจากภาษาดอตเน็ตอื่นๆ สามารถถูกใช้กับโปรแกรมประยุกต์บนเว็บเขียนโปรแกรมสำหรับหน่วยประมวลผลกราฟิกส์ หรือ GPU และใช้เขียนโปรแกรมสำหรับระบบปฏิบัติการแมคโอเอสเท็น ไอโอเอสและ แอนดรอยด์

ปี ค.ศ. 2006มีการพัฒนาโปรแกรม Windows PowerShell สนับสนุนการทำงานของ System Administrator โดยมีลักษณะการทำงานในสภาวะแวดล้อมทั้งแบบ Interactive และ Scripting ซึ่งสามารถที่จะทำงานได้ทั้งแบบ Independently และ Combination

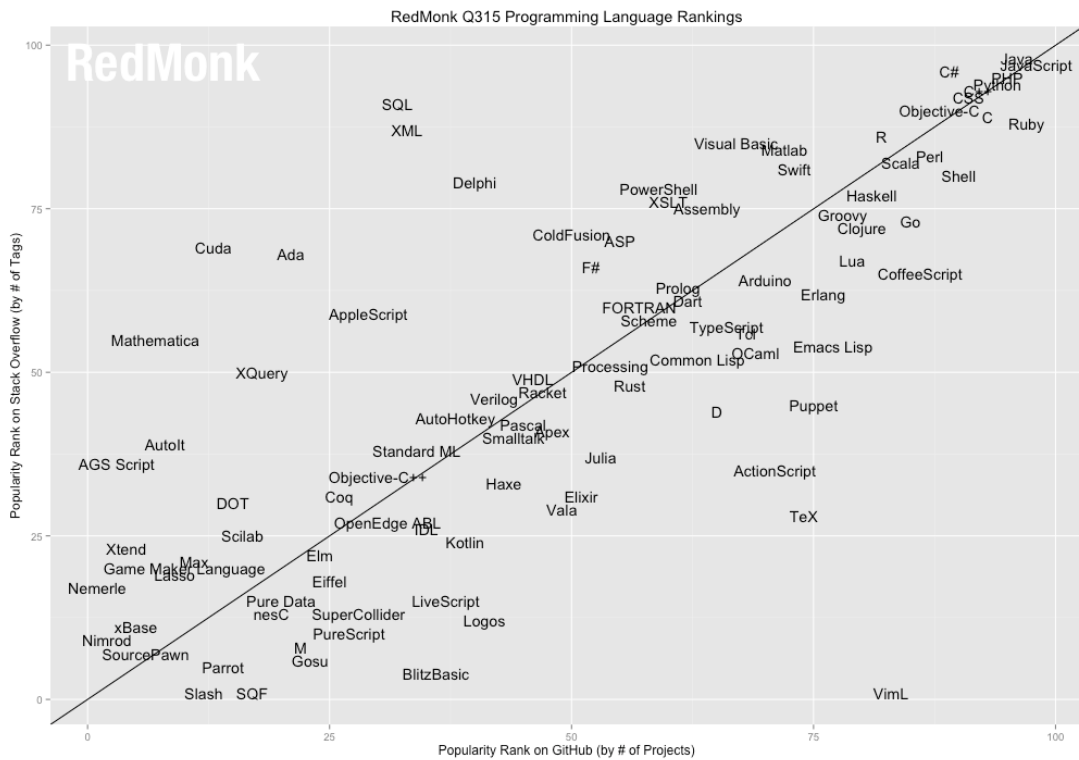
ปี ค.ศ. 2007มีการพัฒนาโปรแกรมภาษา Clojure พัฒนาขึ้นโดย Rich Hickey เป็นภาษาเขียนโปรแกรมมาจากภาษา LISP โดยถูกออกแบบให้เป็นภาษาอเนกประสงค์ที่ Compile เป็น JVM bytecode รันอยู่บน Java Virtual Machine ทำให้สามารถนำ Java Framework มาใช้ใน Clojure ได้เช่นเดียวกับที่ใช้ในภาษา JAVA

ปี ค.ศ. 2009มีการพัฒนาโปรแกรมภาษา Go พัฒนาโดย Google สร้างภาษาโปรแกรมของตัวเองซึ่งเป็นโอเพนซอร์สขึ้นมาเพื่อทำให้โปรแกรมเขียนได้ง่ายขึ้นและเร็วขึ้น ภาษา Go เป็นภาษาที่ปลอดภัยไม่เขียนข้อมูลลงไปในตำแหน่งที่ไม่ควรเขียนในหน่วยความจำและสามารถสนับสนุนการทำงานในหลายกระบวนการของโปรแกรมพร้อมกัน

ปี ค.ศ. 2011มีการพัฒนาโปรแกรมภาษา Dart เป็นภาษาสคริปต์ รองรับการเขียนโปรแกรมแบบเชิงวัตถุมีแนวคิดของ class และ inheritance ส่วนการประกาศชนิดตัวแปรจะประกาศหรือไม่ก็ได้ (optional types) ช่วยให้การสร้างโค้ดขั้นต้นทำได้เร็ว และสามารถมาจัดระเบียบตัวแปรที่หลังได้

ปี ค.ศ. 2014มีการพัฒนาโปรแกรมภาษา Swift ภาษาเขียนโปรแกรมที่ใช้งานง่าย ออกแบบและสร้างสรรค์โดย Apple เพื่อใช้สร้างแอปพลิเคชันสำหรับ iOS, Mac, Apple TV และ Apple Watch ออกแบบมาเพื่อให้นักพัฒนามีอิสระใช้งานง่ายและเป็นโอเพ่นซอร์ส (Opensource)

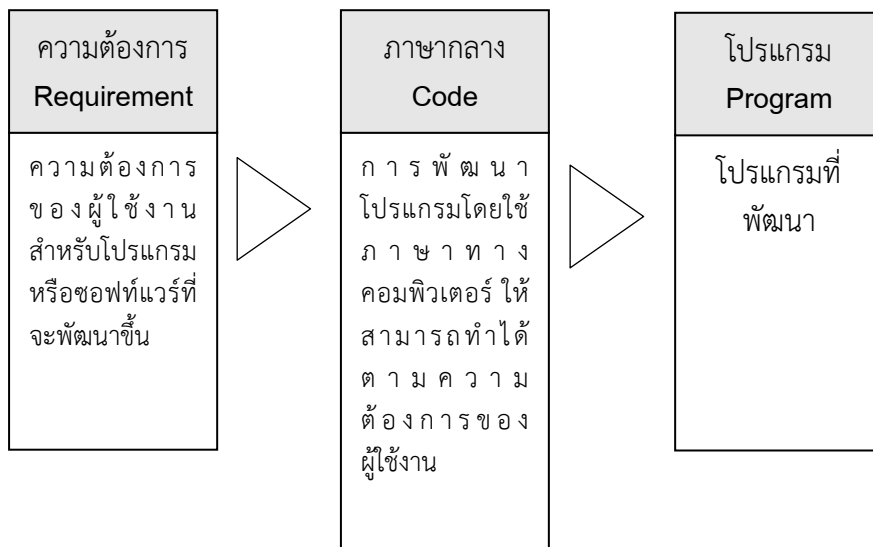
ปี ค.ศ. 2015มีการพัฒนาโปรแกรมภาษา Rust เป็นภาษาโปรแกรมภาษาใหม่ที่พัฒนาโดย Mozilla โดยมุ่งเป้าว่าต้องการเป็นภาษาสำหรับงานประมวลผลที่ทำงานพร้อมๆ กัน (concurrency) และทำงานบนระบบขนาดใหญ่ ออกแบบมาไม่ให้มีปัญหาด้านหน่วยความจำอย่าง buffer overflow การใช้งานจะเป็นระบบคอมไพล์ ตัว syntax ของภาษาจะคล้ายกับ C/C++ คือเป็นภาษาที่ใช้วงเล็บปีกกา {} และประกาศตัวแปรแบบ static



ภาพที่ 1-1 แสดงสถิติการใช้งานโปรแกรมภาษาบนเว็บไซต์GitHub.com
 (ที่มาจากเว็บไซต์ <http://sogrady-media.redmonk.com/sogrady/files/2015/07/lang-rank-615-wm.png>)

1.2 หลักการและแนวคิดการเขียนโปรแกรมคอมพิวเตอร์

การพัฒนาโปรแกรมคอมพิวเตอร์ส่วนมากจุดเริ่มต้นจากความต้องการของผู้ใช้งาน(User requirement) ที่ต้องการคอมพิวเตอร์ไปใช้ในกระบวนการทำงาน ช่วยในการคำนวณที่ซับซ้อน ใช้ในการทำงานที่มีลักษณะเหมือนๆ กัน เพื่อให้ได้ผลลัพธ์อย่างรวดเร็ว แข่งขันกับเวลา และมีความถูกต้องแม่นยำสูงโปรแกรมคอมพิวเตอร์ที่ทำการพัฒนาขึ้นเปรียบเสมือนภาษากลาง(code) ที่ทำให้ผู้พัฒนาโปรแกรม(programmer) สั่งงานให้คอมพิวเตอร์ทำงานหรือดำเนินการโดยมีลำดับขั้นตอนตามโครงสร้างของภาษานั้นๆ ที่ใช้แปลความหมายดังนั้นผู้พัฒนาโปรแกรมต้องเข้าใจหลักการเขียนโปรแกรมคอมพิวเตอร์เป็นอย่างดี เพื่อจะพัฒนาโปรแกรมคอมพิวเตอร์ให้คำนวณผลลัพธ์ได้อย่างถูกต้อง และควรมีคู่มือการพัฒนาโปรแกรมรวมถึงเอกสารอ้างอิงต่างๆ แหล่งค้นหาความรู้เพิ่มเติมเพื่อช่วยการพัฒนาโปรแกรมคอมพิวเตอร์ต่อไป

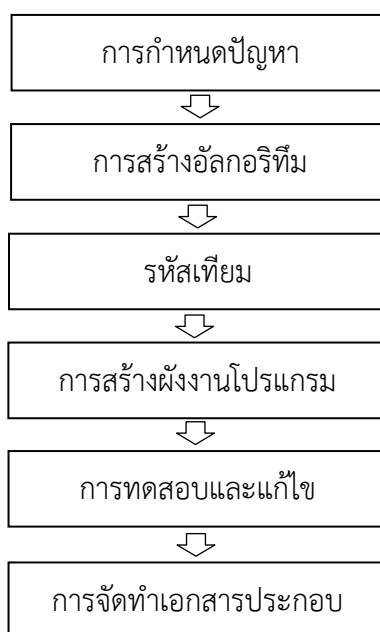


ภาพที่ 1-2แนวคิดการเขียนโปรแกรมคอมพิวเตอร์

1.3 ขั้นตอนการเขียนโปรแกรมคอมพิวเตอร์ (Programming Process)

ขั้นตอนการเขียนโปรแกรมคอมพิวเตอร์ คือ การพัฒนาโปรแกรมคอมพิวเตอร์ที่มีรูปแบบตามลำดับขั้นที่ได้กำหนดไว้แล้วและมีคำอธิบายในแต่ละขั้นตอน โดยมีลำดับขั้นตอนที่สำคัญ 6 ขั้นตอนดังต่อไปนี้

- 1) การกำหนดปัญหา (Problem Definition) คือ เป็นขั้นตอนในการกำหนดว่าโปรแกรมที่จะพัฒนาขึ้นมีผลลัพธ์อะไร มีการใช้ข้อมูลอะไร
- 2) การสร้างอัลกอริทึมและรหัสเทียมคือ การอธิบายขั้นตอนการทำงานของโปรแกรม เพื่อให้ผู้พัฒนาสามารถเขียนโปรแกรมได้
- 3) การสร้างผังงานโปรแกรมเป็นขั้นตอนการเขียนภาพการทำงานของโปรแกรม
- 4) การเขียนโปรแกรมการพัฒนาภาษาคอมพิวเตอร์จากอัลกอริทึมที่ได้กำหนดไว้
- 5) การทดสอบและแก้ไขเป็นขั้นตอนที่ทดสอบหาข้อผิดพลาดของโปรแกรม เมื่อเจอทำการแก้ไขปรับปรุง
- 6) การจัดทำเอกสารประกอบ ผู้พัฒนาจะทำเอกสารในการพัฒนาโปรแกรม ขั้นตอนการพัฒนา ซอฟต์แวร์ที่ใช้ เพื่อเมื่อต้องการพัฒนาใหม่ หรือผู้ที่สนใจมาอ่าน



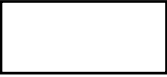
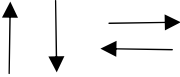
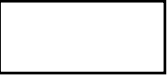
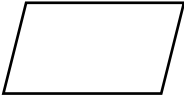
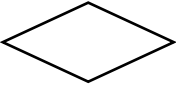


ภาพที่ 1-3 ขั้นตอนการเขียนโปรแกรมคอมพิวเตอร์

1.4 ผังงานการเขียนโปรแกรม



ผังงานการเขียนโปรแกรม คือ การนำอัลกอริทึมหรือรหัสเทียมที่ได้สร้างไว้ มาแปลงเป็นสัญลักษณ์สากล

สัญลักษณ์ที่ใช้ในการเขียนผังงาน กำหนดตามสถาบันแห่งชาติอเมริกัน (ANSI : American National Standards Institute) ตัวอย่างสัญลักษณ์และความหมายที่ใช้กันหลักๆ มีดังนี้

ตารางที่ 1 สัญลักษณ์การเขียนผังงาน

สัญลักษณ์	คำอธิบาย
	สัญลักษณ์ terminal หมายถึงจุดเริ่มต้น / สิ้นสุดของโปรแกรม
	สัญลักษณ์arrow หมายถึงลูกศรแทนการแสดงลำดับทิศทางการทำงานของโปรแกรมและการไหลของข้อมูล
	สัญลักษณ์processing หมายถึง การประมวลผล ใช้สำหรับแสดงคำสั่งในการประมวลผล หรือการกำหนดค่าข้อมูลให้กับตัวแปร
	สัญลักษณ์ input หรือ output แสดงการอ่านข้อมูลจากหน่วยเก็บข้อมูลสำรองเข้าสู่หน่วยความจำหลักภายในเครื่องหรือการแสดงผลลัพธ์จากการประมวลผลออกมา
	สัญลักษณ์ decision หมายถึงการตรวจสอบเงื่อนไขเพื่อตัดสินใจ โดยจะมีเส้นออกจากรูปเพื่อแสดงทิศทางการทำงานต่อไป เงื่อนไขเป็นจริงหรือเป็นเท็จ
	สัญลักษณ์ document เป็นการแสดงผลหรือรายงานที่ถูกสร้างออกมา
	สัญลักษณ์ on page connector แสดงจุดเชื่อมต่อของผังงานภายใน หรือเป็นที่บรรจบของเส้นหลายเส้นที่มาจากหลายทิศทางเพื่อจะไปสู่การทำงานอย่างใดอย่างหนึ่งที่เหมือนกัน

ตารางที่ 1 สัญลักษณ์การเขียนผังงาน (ต่อ)

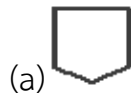
สัญลักษณ์	คำอธิบาย
	สัญลักษณ์ off page connector การขึ้นหน้าใหม่ ในกรณีที่ผังงานมีความยาวเกินกว่าที่จะแสดงพอในหนึ่งหน้า
	สัญลักษณ์ display แทนการแสดงผลทางจอภาพ

บทสรุป

เนื้อหาในบทนี้ผู้เรียนสามารถ เป็นบทนำในการเขียนโปรแกรมคอมพิวเตอร์โดยให้ทราบประวัติการเขียนโปรแกรมคอมพิวเตอร์ว่าเริ่มตั้งแต่ ค.ศ. ไต มีภาษาอะไรบ้างที่พัฒนาขึ้น ภาษาใดมีการพัฒนาจนถึงยุคปัจจุบันหลักการและแนวคิดการเขียนโปรแกรมคอมพิวเตอร์ เพื่อให้พัฒนาโปรแกรมได้อย่างถูกต้อง ขั้นตอนการเขียนโปรแกรมคอมพิวเตอร์ ว่ามีจำนวนกี่ขั้นตอน อะไรบ้าง รวมถึงการเขียนผังงานโปรแกรมคอมพิวเตอร์ เป็นการทบทวนข้อมูลที่ได้เรียนมาแล้ว หรือมีความรู้อยู่แล้ว ซึ่งสามารถนำความรู้ที่ได้ไปพัฒนาสร้างโปรแกรม และผังงานโปรแกรมคอมพิวเตอร์ได้ต่อไป

แบบฝึกหัดท้ายบท

- 1) จงอธิบายหลักการเขียนโปรแกรม พร้อมยกตัวอย่างโปรแกรมประกอบ
- 2) จงอธิบายขั้นตอนการเขียนโปรแกรม
- 3) ให้ยกตัวอย่างการเขียนผังงานโปรแกรมที่ใช้ในชีวิตประจำวัน
- 4) จงเสนอแนวคิดผังงานโปรแกรมเกมคอมพิวเตอร์
- 5) จงบอกประโยชน์ของการเขียนผังงานโปรแกรม
- 6) จงอธิบายความหมายของสัญลักษณ์ของการเขียนผังงานโปรแกรม พร้อมยกตัวอย่างประกอบ ดังต่อไปนี้



เอกสารอ้างอิง

- Curtis Bennett. (2009). **A Simple Introduction to Game Programming With C# and XNA 3.1**. 1. United States of America :xnagamemaking.com.
- มนตรี พจนารถาวัดณ์. (2535). **การเขียนโปรแกรมคอมพิวเตอร์ด้วยเทอร์โบซี**. กรุงเทพมหานคร, ซีเอ็ดยูเคชั่น.
- ชนทรศรี พลเดช. (2557). **การเขียนโปรแกรมคอมพิวเตอร์และอัลกอริทึม cssc1101**. สาขาวิชาวิทยาการคอมพิวเตอร์. มหาวิทยาลัยราชภัฏจันทรเกษม.
- ทัศนีย์ รอดมันคง. (2559). **การโปรแกรมสำหรับมัลติมีเดีย**. คณะวิทยาศาสตร์. มหาวิทยาลัยราชภัฏจันทรเกษม.
- _____. (2559). History of programming languages. [ออนไลน์]. เข้าถึงข้อมูลวันที่ 2 พฤษภาคม 2559 จาก https://en.wikipedia.org/wiki/History_of_programming_languages
- Knuth, Donald E.; Pardo, Luis Trabb. **Early development of programming languages**. Encyclopedia of Computer Science and Technology. Marcel Dekker.
- J. Fuegi and J. Francis (October–December 2003), **Lovelace & Babbage and the creation of the 1843 'notes'**, Annals of the History of Computing, 25 (4).
- Rojas, Raúl, et al. (2000). **Plankalkül: The First High-Level Programming Language and its Implementation**. Institut frame Informatik, Freie Universität Berlin, Technical Report B-3/2000.
- Eugene Loh (18 June 2010). **The Ideal HPC Programming Language**. Queue. Association of Computing Machines.
- _____. (2015). **HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers**.
- Rojas, Raúl; Hashagen, Ulf (2002). **The First Computers: History and Architectures**. MIT Press. p. 292. ISBN 978-0262681377. Retrieved October 25, 2013.

