

## คำนำ

เอกสารคำสอนนี้จัดทำขึ้นด้วยวัตถุประสงค์เพื่อใช้เป็นคู่มือการเรียน สำหรับนักศึกษาชั้นปีที่สองสาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ต้องเรียนรายวิชา 423206 หลักการของภาษาการทำให้โปรแกรมเป็นรายวิชาบังคับในหลักสูตรวิศวกรรมศาสตรบัณฑิตสาขาวิชาวิศวกรรมคอมพิวเตอร์ ตามแผนการศึกษานักศึกษาจะเรียนรายวิชานี้ในภาคการศึกษาที่ 6 (ในระบบไตรภาค) ดังนั้นนักศึกษาจะมีพื้นฐานความรู้ในเรื่องการทำให้โปรแกรมคอมพิวเตอร์เบื้องต้น (เน้นภาษา C) และการทำให้โปรแกรมเชิงวัตถุ (เน้นภาษา Java) ดังนั้นรายวิชานี้จึงมุ่งเน้นให้นักศึกษาเข้าใจถึงแนวคิดและหลักการของภาษาการทำให้โปรแกรมโดยใช้ภาษา C/C++ และ Java เป็นหลักในการอธิบาย ภาษาอื่นๆ เช่น Pascal, Ada, FORTRAN จะใช้เป็นตัวอย่างเสริม นอกจากการทำให้โปรแกรมเชิงคำสั่งและเชิงวัตถุแล้ว ในรายวิชานี้ยังแนะนำผู้เรียนให้รู้จักอีกสองแนวทางที่เป็นที่นิยมในการทำให้โปรแกรมประยุกต์นั่นคือ การทำให้โปรแกรมเชิงหน้าที่ (หรือเชิงฟังก์ชัน) และการทำให้โปรแกรมเชิงตรรกะ ภาษาคอมพิวเตอร์ที่ใช้ประกอบการอธิบายคือ ภาษา Scheme และ ภาษา Prolog นอกจากภาษาที่ใช้ในการทำแอปพลิเคชันต่างๆ แล้วนักศึกษายังจะได้เรียนเกี่ยวกับภาษาฟอร์มอลที่ใช้เป็นภาษาเมตาเพื่อการอธิบายความหมายของภาษาคอมพิวเตอร์ รวมถึงรู้จักการพิสูจน์เพื่อยืนยันความถูกต้องของโปรแกรม

## ประมวลรายวิชา (Syllabus)

1. รหัสและชื่อวิชา 423206 Principles of Programming Languages
2. จำนวนหน่วยกิต 4 หน่วยกิต (บรรยาย 4 ชั่วโมง/สัปดาห์)
3. วิชาระดับ ปริญญาตรี สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
4. สถานภาพของวิชา วิชาบังคับ
5. ชื่อผู้สอน ดร.นิตยา เกิดประสพ

### 6. เนื้อหารายวิชา (course description)

ข้อมูลและการควบคุมเชิงนามธรรม; การกำหนดคุณลักษณะการตรวจชนิดโครงสร้าง;  
วิธีเชิงแบบในการอธิบายโปรแกรม; รูปแบบเชิงแผนและความหมายเชิงแบบ; วิธีการ  
พิสูจน์ความถูกต้องของโปรแกรม

### 7. หัวข้อการสอน (course outline)

- (1) Programming language concepts
- (2) Syntax of programming languages
- (3) Semantics of programming languages
- (4) Language translation and processing
- (5) Data types
- (6) Program control structures
- (7) Imperative programming languages
- (8) Object-oriented programming languages
- (9) Functional programming languages
- (10) Logic programming language

### 8. การวัดผลการเรียน

คะแนนเต็ม 100 คะแนน แบ่งเป็น

สอบกลางภาค	30 คะแนน
การมีส่วนร่วมในชั้นเรียน	10 คะแนน
โปรเจ็คและการบ้าน	20 คะแนน
สอบปลายภาค	40 คะแนน

การทำการเลือกโปรเจ็ค ให้เลือกจากหัวข้อต่อไปนี้

- Visual languages
- Script languages

- Network programming
- Parallel programming
- Aspect-oriented programming
- Event-driven programming

โปรเจ็คส่งเป็นรายงาน (20-40 หน้า) อธิบาย

- ความหมาย / หลักการ / ลักษณะเด่น
- ข้อแตกต่างจาก C หรือ Java
- ตัวอย่าง

## 9. รายชื่อหนังสืออ่านประกอบ

- (1) นิติยา เกิดประสพ. *เอกสารคำสอนรายวิชา หลักการของภาษาการทำให้โปรแกรม*. มหาวิทยาลัยเทคโนโลยีสุรนารี, 2547.
- (2) Robert W. Sebesta. *Concepts of Programming Languages*. Addison-Wesley, 2001.
- (3) Carlo Ghezzi and Mehdi Jazayeri. *Programming Language Concepts*. John Wiley & Sons, 1997.
- (4) Allen Tucker and Robert Noonan. *Programming Languages: Principles and Paradigms*. McGraw-Hill, 2002
- (5) Terrence W. Pratt and Marvin V. Zelkowitz. *Programming Languages: Design and Implementation*, fourth edition. Prentice Hall, 2001.

## แผนการสอน

สัปดาห์ที่	เนื้อหาการเรียน	กิจกรรม
1	บทที่ 1 แนวคิดของภาษาการทำให้โปรแกรม <ul style="list-style-type: none"> <li>• ความสัมพันธ์ของภาษาและวัฏจักรการพัฒนาซอฟต์แวร์</li> <li>• การพิจารณาคุณลักษณะของภาษา</li> <li>• แนวทางการจัดกลุ่มภาษา</li> </ul>	<ul style="list-style-type: none"> <li>▶ นักศึกษาอ่านเอกสารทบทวนประวัติของภาษาคอมพิวเตอร์</li> <li>▶ ทำแบบฝึกหัดท้ายบทที่ 1</li> </ul>
2	บทที่ 2 ไวยากรณ์ของภาษาการทำให้โปรแกรม <ul style="list-style-type: none"> <li>• เกณฑ์การออกแบบและส่วนประกอบพื้นฐานของไวยากรณ์ของภาษา</li> <li>• รูปแบบทางการในการกำหนดไวยากรณ์</li> <li>• การวิเคราะห์ไวยากรณ์และการจัดกลุ่มไวยากรณ์ตามลำดับชั้นขอมสกี</li> </ul>	<ul style="list-style-type: none"> <li>▶ นักศึกษาอ่านเอกสารคำสอนบทที่ 2</li> <li>▶ ทำแบบฝึกหัดท้ายบทที่ 2</li> </ul>
3	บทที่ 3 ความหมายของภาษาการทำให้โปรแกรม <ul style="list-style-type: none"> <li>• การอธิบายความหมายเชิงดำเนินการ</li> <li>• การอธิบายความหมายเชิงพิสูจน์</li> <li>• การอธิบายความหมายเชิงแทนความ</li> <li>• การตรวจสอบความถูกต้องของโปรแกรม</li> </ul>	<ul style="list-style-type: none"> <li>▶ นักศึกษาอ่านเอกสารคำสอนบทที่ 3 ก่อนเข้าชั้นเรียน</li> <li>▶ ทำแบบทดสอบในชั้นเรียน</li> <li>▶ การบ้านการพิสูจน์โปรแกรม</li> </ul>
4	บทที่ 4 การแปลภาษาและการประมวลผล <ul style="list-style-type: none"> <li>• อินเตอร์พรีเตอร์และคอมไพเลอร์</li> <li>• ขั้นตอนการทำงานของคอมไพเลอร์</li> <li>• เครื่องจักรเสมือน</li> <li>• โครงสร้างหน่วยความจำขณะประมวลผล</li> </ul>	<ul style="list-style-type: none"> <li>▶ นักศึกษาอ่านเอกสารคำสอนบทที่ 4</li> <li>▶ ทำแบบฝึกหัดท้ายบทที่ 4</li> </ul>
5	บทที่ 5 แบบชนิดข้อมูล <ul style="list-style-type: none"> <li>• คุณสมบัติของข้อมูล</li> <li>• ระบบชนิดข้อมูล</li> <li>• ชนิดข้อมูลแบบสเกลาร์ แบบคอมโพสิต และแบบนามธรรม</li> </ul>	<ul style="list-style-type: none"> <li>▶ นักศึกษาอ่านเอกสารคำสอนบทที่ 5</li> <li>▶ ทำแบบฝึกหัดท้ายบทที่ 5</li> </ul>
6	บทที่ 6 โครงสร้างการควบคุมการทำงานของโปรแกรม	▶ นักศึกษาอ่านเอกสารคำสอนบทที่ 6

	<ul style="list-style-type: none"> <li>• นิพจน์และคำสั่งกำหนดค่า</li> <li>• คำสั่งเป็นกลุ่ม คำสั่งแบบมีเงื่อนไข และคำสั่งทำซ้ำ</li> <li>• โปรแกรมย่อย</li> <li>• การจัดการกรณียกเว้น</li> </ul>	▶ ทำแบบฝึกหัดท้ายบทที่ 6
7	ทบทวนเนื้อหาในบทที่ 1 – 6 <b>สอบกลางภาค</b>	
8	บทที่ 7 การทำโปรแกรมเชิงคำสั่ง <ul style="list-style-type: none"> <li>• หลักการพื้นฐาน</li> <li>• ตัวแปรและชนิดข้อมูล</li> <li>• การควบคุมลำดับการทำงาน</li> </ul>	▶ นักศึกษาอ่านเอกสารคำสอนบทที่ 7 ▶ ทำแบบฝึกหัดท้ายบทที่ 7 ▶ การบ้านการทำโปรแกรมเชิงคำสั่ง
9	บทที่ 8 การทำโปรแกรมเชิงวัตถุ <ul style="list-style-type: none"> <li>• หลักการพื้นฐาน</li> <li>• การรวมข้อมูลเป็นหน่วยเดียว</li> <li>• การถ่ายทอด</li> <li>• การมีหลายรูป</li> </ul>	▶ นักศึกษาอ่านเอกสารคำสอนบทที่ 8 ▶ ทำแบบฝึกหัดท้ายบทที่ 8 ▶ การบ้านการทำโปรแกรมเชิงวัตถุ
10	บทที่ 9 การทำโปรแกรมเชิงหน้าที่ <ul style="list-style-type: none"> <li>• หลักการพื้นฐาน</li> <li>• แคลคูลัสแลมบ์ดา</li> <li>• ภาษา LISP และ Scheme</li> <li>• ชนิดข้อมูลลิสต์และการประมวลผลลิสต์</li> <li>• ฟังก์ชันเวียนบังเกิดและฟังก์ชันอันดับสูง</li> </ul>	▶ นักศึกษาอ่านเอกสารคำสอนบทที่ 9 ▶ ทำแบบฝึกหัดท้ายบทที่ 9 ▶ การบ้านการทำโปรแกรมเชิงหน้าที่
11	บทที่ 10 การทำโปรแกรมเชิงตรรกะ <ul style="list-style-type: none"> <li>• หลักการพื้นฐาน</li> <li>• พื้นฐานตรรกศาสตร์</li> <li>• เรโซลูชันและยูนิฟิเคชัน</li> <li>• ภาษา Prolog</li> <li>• การทำโปรแกรมในภาษา Prolog</li> </ul>	▶ นักศึกษาอ่านเอกสารคำสอนบทที่ 10 ▶ ทำแบบฝึกหัดท้ายบทที่ 10 ▶ การบ้านการทำโปรแกรมเชิงตรรกะ
12	• ทบทวนเนื้อหาบทที่ 7-10	▶ นักศึกษาส่งโปรเจ็ค
13	<b>สอบปลายภาค</b>	

## สารบัญ

	หน้า
คำนำ	i
ประมวลรายวิชาและแผนการสอน	ii
สารบัญ	vi
สารบัญภาพ	x
บทที่ 1 แนวคิดของภาษาการทำให้โปรแกรม (Programming language concepts)	1
1.1 ความสัมพันธ์ของภาษาและวัฏจักรการพัฒนาซอฟต์แวร์ (Relationship of language and software development cycle)	2
1.2 การพิจารณาคูณลักษณะของภาษา(Language property criteria)	3
1.3 แนวทางการจัดกลุ่มภาษา (Programming language paradigms)	7
1.4 ประวัติของภาษาคอมพิวเตอร์ (History of programming languages)	9
1.5 สรุป	18
แบบฝึกหัดท้ายบทที่ 1	19
บทที่ 2 ไวยากรณ์ของภาษาการทำให้โปรแกรม (Syntax of programming languages)	21
2.1 เกณฑ์การออกแบบไวยากรณ์ของภาษา (Syntactic design criteria)	22
2.2 ส่วนประกอบพื้นฐานของไวยากรณ์ภาษา (Syntactic elements of a language)	24
2.3 รูปแบบทางการในการกำหนดไวยากรณ์ของภาษา (Formal methods to syntax definition)	27
2.4 การวิเคราะห์ไวยากรณ์ (Syntactic analysis)	32
2.5 ไวยากรณ์กำกวม (Ambiguous grammar)	39
2.6 การจัดกลุ่มไวยากรณ์ตามลำดับชั้นชอมสกี (Class of grammars by Chomsky hierarchy)	41
2.7 เครื่องเชิงนามธรรม (Abstract machines)	43
2.8 สรุป	46
แบบฝึกหัดท้ายบทที่ 2	47
บทที่ 3 ความหมายของภาษาการทำให้โปรแกรม (Semantics of programming languages)	50
3.1 ไวยากรณ์เชิงลักษณะประจำ (Attribute grammar)	51
3.2 การอธิบายความหมายเชิงดำเนินการ (Operational semantics)	52

3.3	การอธิบายความหมายเชิงพิสูจน์ (Axiomatic semantics)	56
3.4	การอธิบายความหมายเชิงแทนความ (Denotational semantics)	60
3.5	การตรวจสอบความถูกต้องของโปรแกรม (Program verification)	63
3.6	สรุป	70
	แบบฝึกหัดท้ายบทที่ 3	71
บทที่ 4	การแปลภาษาและการประมวลผล (Language translation and processing)	72
4.1	อินเตอร์พรีเตอร์และคอมไพเลอร์ (Interpreter and compiler)	73
4.2	ขั้นตอนการทำงานของคอมไพเลอร์ (Stages in compilation)	78
4.3	เครื่องจักรเสมือน (Virtual machine)	87
4.4	โครงสร้างหน่วยความจำขณะประมวลผล (Run-time memory structure)	91
4.5	สรุป	112
	แบบฝึกหัดท้ายบทที่ 4	113
บทที่ 5	แบบชนิดข้อมูล (Data types)	116
5.1	คุณสมบัติของข้อมูล (Properties of data)	117
5.2	ระบบชนิดข้อมูล (Type system)	118
5.3	ชนิดข้อมูลแบบสเกลาร์ (Scalar data type)	124
5.4	ชนิดข้อมูลแบบคอมโพสิต (Composite data type)	127
5.5	ชนิดข้อมูลนามธรรม (Abstract data type)	139
5.6	สรุป	141
	แบบฝึกหัดท้ายบทที่ 5	142
บทที่ 6	โครงสร้างการควบคุมการทำงานของโปรแกรม (Program control structures)	144
6.1	นิพจน์ (Expression)	145
6.2	คำสั่งกำหนดค่า (Assignment statement)	150
6.3	คำสั่งเป็นกลุ่ม (Compound statement)	151
6.4	คำสั่งแบบมีเงื่อนไข (Conditional statement)	152
6.5	คำสั่งทำซ้ำ (Iteration statement)	153
6.6	โปรแกรมย่อย (Subprogram)	156
6.7	การจัดการกรณียกเว้น (Exception handling)	157

6.8	สรุป	158
	แบบฝึกหัดท้ายบทที่ 6	159
บทที่ 7	การทำโปรแกรมเชิงคำสั่ง (Imperative programming)	162
7.1	หลักการทำโปรแกรมเชิงคำสั่ง (Principles of imperative programming)	163
7.2	ตัวแปรและชนิดข้อมูล (Variables and types)	165
7.3	การควบคุมลำดับการทำงาน (Flow of control)	168
7.4	ส่วนประกอบของโปรแกรม (Program composition)	169
7.5	การทำโปรแกรมเชิงคำสั่งในภาษาซี, ปาสคาล, เอดา และฟอร์แทรน (Imperative programming in C, Pascal, Ada and FORTRAN)	176
7.6	สรุป	179
	แบบฝึกหัดท้ายบทที่ 7	180
บทที่ 8	การทำโปรแกรมเชิงวัตถุ (Object-oriented programming)	184
8.1	หลักการทำโปรแกรมเชิงวัตถุ (Principles of object-oriented programming)	185
8.2	การรวมข้อมูลเป็นหน่วยเดียว (Data encapsulation)	188
8.3	การถ่ายทอด (Inheritance)	190
8.4	การมีหลายรูปและการยึดเหนี่ยวแบบพลวัต (Polymorphism and dynamic binding)	192
8.5	การทำโปรแกรมเชิงวัตถุในภาษาจาวา, ซีพลัสพลัส และ สมอลทอล์ค (Object-oriented programming in Java, C++ and Smalltalk)	194
8.6	สรุป	197
	แบบฝึกหัดท้ายบทที่ 8	198
บทที่ 9	การทำโปรแกรมเชิงหน้าที่ (Functional programming)	203
9.1	การทำโปรแกรมเชิงกระบวนการคำสั่งและไร้กระบวนการคำสั่ง (Procedural and non-procedural programming)	204
9.2	หลักการของการทำโปรแกรมเชิงหน้าที่ (Principles of functional programming)	206
9.3	ฟังก์ชัน (Function)	208
9.4	แคลคูลัสแลมบ์ดา (Lambda calculus)	211
9.5	ภาษาลิสป์และสคีม (LISP and Scheme languages)	216



9.6 โปรแกรมและนิพจน์ (Program and expression)	218
9.7 ชนิดข้อมูลลิสต์และการประมวลผลลิสต์ (List and list processing)	222
9.8 ฟังก์ชันเวียนบังเกิด (Recursive function)	225
9.9 ฟังก์ชันอันดับสูง (Higher-order function)	228
9.10 การทำโปรแกรมเชิงหน้าที่ในภาษาลิสป์และสคิม (Functional programming in LISP and Scheme)	230
9.11 สรุป	233
แบบฝึกหัดท้ายบทที่ 9	234
บทที่ 10 การทำโปรแกรมเชิงตรรกะ (Logic programming)	238
10.1 หลักการทำโปรแกรมเชิงตรรกะ (Principles of logic programming)	239
10.2 พื้นฐานตรรกศาสตร์ (Fundamentals of logic)	241
10.3 เรโซลูชันและยูนิฟิเคชัน (Resolution and unification)	248
10.4 หลักการพื้นฐานของภาษาโปรล็อก (Fundamentals of Prolog)	252
10.5 การทำโปรแกรมเชิงตรรกะในภาษาโปรล็อก (Logic programming in Prolog)	268
10.6 การทำโปรแกรมเชิงตรรกะแบบมีเงื่อนไขบังคับ (Constraint logic programming)	272
10.7 สรุป	274
แบบฝึกหัดท้ายบทที่ 10	275
บรรณานุกรม	279

## สารบัญภาพ

	หน้า
รูปที่ 1.1 โปรแกรมภาษาเครื่องคำนวณตัวหารร่วมมาก	9
รูปที่ 1.2 โปรแกรมภาษาแอสเซมบลีคำนวณตัวหารร่วมมาก	9
รูปที่ 1.3 จุดกำเนิดของภาษาคอมพิวเตอร์ระดับสูงตั้งแต่ยุคเริ่มต้นจนถึงปัจจุบัน	11
รูปที่ 1.4 เส้นเวลาแสดงจุดกำเนิดของภาษาคอมพิวเตอร์ จำแนกตามกลุ่มการทำโปรแกรม	12
รูปที่ 2.1 ความหมายสองรูปแบบของคำสั่งแบบมีเงื่อนไข	23
รูปที่ 2.2 เป็นพิมพ์พิเศษที่ใช้สำหรับพิมพ์คำสั่งในภาษา APL	24
รูปที่ 2.3 ขั้นตอนการทำงานของคอมไพเลอร์	32
รูปที่ 2.4 ไวยากรณ์ของภาษาตัวอย่างโดยแสดงเฉพาะส่วนคำศัพท์	33
รูปที่ 2.5 ไวยากรณ์ของภาษาตัวอย่างส่วนโปรแกรมและข้อความสั่ง	34
รูปที่ 2.6 โปรแกรมหาเลขไฟโบนาชชีตัวที่ $n$ โดย $n = 8$	35
รูปที่ 2.7 การจัดกลุ่มโทเคนจากข้อความ สามบรรทัดแรกของโปรแกรมในรูปที่ 2.6	35
รูปที่ 2.8 พาสทรีของการวิเคราะห์ไวยากรณ์โปรแกรมตัวอย่าง	37
รูปที่ 2.9 พาสทรีสองรูปแบบที่ใช้ตรวจสอบนิพจน์ 2-3-4	39
รูปที่ 2.10 พาสทรีสองรูปแบบของคำสั่ง IF_THEN_ELSE	40
รูปที่ 2.11 เครื่องอัตโนมัติสถานะจำกัดที่ทำหน้าที่ตรวจสอบ SignedIntegers	43
รูปที่ 2.12 ภาพเปรียบเทียบเครื่องเชิงนามธรรมทั้งสี่ประเภท	45
รูปที่ 3.1 พาสทรีที่ตรวจสอบไวยากรณ์ของนิพจน์และแสดงการคำนวณค่าของนิพจน์	51
รูปที่ 4.1 โครงสร้างการทำงานของอินเตอร์พรีเตอร์	73
รูปที่ 4.2 เปรียบเทียบวัฏจักรการทำงานของอินเตอร์พรีเตอร์และซีพียู	74
รูปที่ 4.3 โครงสร้างการทำงานของคอมไพเลอร์	75
รูปที่ 4.4 การแปลแบบแยกส่วนของคอมไพเลอร์	76
รูปที่ 4.5 โครงสร้างแสดงขั้นตอนการทำงานของคอมไพเลอร์	78
รูปที่ 4.6 โปรแกรมค้นหาทางในภาษาปาสคาล	79
รูปที่ 4.7 สายของโทเคนจากการวิเคราะห์ศัพท์ของโปรแกรมค้นหาในรูปที่ 4.6	79
รูปที่ 4.8 เครื่องจักรสถานะจำกัดแสดงการตรวจจับคำศัพท์ (บางส่วน) ของภาษาปาสคาล	80
รูปที่ 4.9 ขั้นตอนวิธีของสแกนเนอร์ที่ตรวจจับคำศัพท์ (บางส่วน) ของภาษาปาสคาล	80
รูปที่ 4.10 โครงสร้างโปรแกรมสแกนเนอร์ที่เขียนด้วยภาษา Java	81
รูปที่ 4.11 โปรแกรมช่วยสร้างคอมไพเลอร์ส่วนเล็กชิคคอลนาไลเซอร์	82
รูปที่ 4.12 โครงสร้างพาสทรีเพื่อตรวจสอบนิพจน์ $x+2*y$	83

รูปที่ 4.13 โครงสร้างของโปรแกรมพาสเชอร์ที่วิเคราะห์ข้อความสั่งกำหนดค่า	84
รูปที่ 4.14 ฟังก์ชัน assignment ในโปรแกรมพาสเชอร์	84
รูปที่ 4.15 โปรแกรมช่วยสร้างคอมไพเลอร์ส่วนพาสเชอร์	85
รูปที่ 4.16 ขั้นตอนการทำงานของคอมไพเลอร์ที่สร้างรหัสเป้าหมายบนเครื่อง คอมพิวเตอร์เหมือน	87
รูปที่ 4.17 ลำดับชั้นของเครื่องจักรเหมือนในการทำงานกับโปรแกรมเว็บ	88
รูปที่ 4.18 ขั้นตอนการประมวลผลโปรแกรมภาษา Java	89
รูปที่ 4.19 ตัวอย่างโปรแกรมในรูปแบบภาษา Java และรูปแบบไบต์โค้ด	89
รูปที่ 4.20 โครงสร้าง Java Runtime Environment (JRE)	90
รูปที่ 4.21 โครงสร้างหน่วยความจำขณะมีการประมวลผลโปรแกรม	91
รูปที่ 4.22 โครงสร้างของสแตกเฟรม	93
รูปที่ 4.23 โครงสร้างหน่วยความจำขณะฟังก์ชัน main ทำงานและขณะมีการเรียกใช้ ฟังก์ชัน A และ B	94
รูปที่ 4.24 โครงสร้างหน่วยความจำเมื่อมีการส่งผ่านพารามิเตอร์ด้วยวิธีส่งโดยอ้างอิง	96
รูปที่ 4.25 โครงสร้างของลิงค์ลิสต์ที่ประกอบขึ้นจาก Node	99
รูปที่ 4.26 การเกิด garbage จากการเปลี่ยนตำแหน่งชี้ของพอยเตอร์	100
รูปที่ 4.27 การเกิด dangling reference จากการสั่งยกเลิกการใช้งานเนื้อที่ในหน่วยความจำ	101
รูปที่ 4.28 โครงสร้างหน่วยความจำที่จัดสรรให้กับข้อมูลอาร์เรย์หนึ่งมิติ	102
รูปที่ 4.29 โครงสร้างหน่วยความจำที่จัดสรรให้กับข้อมูลอาร์เรย์สองมิติ	103
รูปที่ 4.30 โครงสร้างหน่วยความจำของข้อมูลเรคคอร์ด	104
รูปที่ 4.31 โครงสร้างหน่วยความจำฮีพ ก่อนและหลังการใช้คำสั่ง new(5)	104
รูปที่ 4.32 การจัดสรรหน่วยความจำฮีพให้กับอาร์เรย์หนึ่งมิติ	105
รูปที่ 4.33 การจัดสรรหน่วยความจำฮีพให้กับโครงสร้างข้อมูล struct	105
รูปที่ 4.34 ตัวอย่างการเกิด garbage และ dangling reference	106
รูปที่ 4.35 ตัวอย่างโครงสร้างฮีพเมื่อใช้การจัดเก็บขยะด้วยวิธีการนับตัวชี้	107
รูปที่ 4.36 การเกิดขยะโดยจำนวนตัวชี้มีค่ามากกว่าศูนย์	108
รูปที่ 4.37 ตัวอย่างโครงสร้างบล็อกในฮีพ เมื่อใช้การจัดเก็บขยะด้วยวิธีทำเครื่องหมาย	108
รูปที่ 4.38 โครงสร้างบล็อกในฮีพ หลังเสร็จสิ้นช่วงทำเครื่องหมาย	109
รูปที่ 4.39 โครงสร้างในฮีพหลังเสร็จสิ้นช่วงกวาดขยะ	109
รูปที่ 4.40 โครงสร้างเริ่มต้นของฮีพก่อนการจัดเก็บขยะด้วยวิธีการรวบรวมสำเนา	110
รูปที่ 4.41 โครงสร้างฮีพหลังการจัดเก็บขยะด้วยวิธีการรวบรวมสำเนา	111
รูปที่ 5.1 รูปแบบการจัดเก็บข้อมูลเลขจำนวนเต็มในคอมพิวเตอร์	124

รูปที่ 5.2 การจัดเก็บชนิดข้อมูลเลขจำนวนจริงตามมาตรฐาน IEEE 754	125
รูปที่ 5.3 การระบุช่วงข้อมูลบนอาร์เรย์ด้วยคำสั่งภาษาฟอร์แทรน A(1:4,2), B(3,1:3) และ C(3,1:3,1:4)	128
รูปที่ 5.4 โครงสร้างการจัดเก็บข้อมูลของ struct Employee	132
รูปที่ 5.5 โครงสร้างข้อมูลพนักงานที่เรคคอร์ดมีความยาวแปรผัน	133
รูปที่ 5.6 โครงสร้างภายในหน่วยความจำเมื่อมีการใช้ตัวแปรชนิดพอยเตอร์	134
รูปที่ 5.7 การอ้างอิงค่าของตัวแปรพอยเตอร์ไปยังค่าประเภท int, char และ float	134
รูปที่ 5.8 การเพิ่มค่าตัวแปรพอยเตอร์เพื่อชี้ค่า int และ float	136
รูปที่ 5.9 เปรียบเทียบตัวแปรชนิด pointer และชนิด reference	138
รูปที่ 6.1 ลำดับการมาก่อนของตัวดำเนินการในภาษา C	146
รูปที่ 6.2 ตัวอย่างเปรียบเทียบวิธีการเขียนนิพจน์ในรูปแบบอินฟิกซ์, พรีฟิกซ์ และโพสต์ฟิกซ์	147
รูปที่ 6.3 โครงสร้างภายในหน่วยความจำของตัวแปร x, y, i	150
รูปที่ 6.4 โครงสร้างภายในหน่วยความจำภายหลังทำคำสั่ง $x = y * i$	151
รูปที่ 7.1 ความสัมพันธ์ของชื่อตัวแปร, ค่าของข้อมูลและโครงสร้างที่เกี่ยวข้องเมื่อมีการประกาศตัวแปร <code>int i=5</code>	165
รูปที่ 7.2 การแทนเลขจำนวนจริงตามมาตรฐาน IEEE 754	166
รูปที่ 7.3 ชนิดข้อมูลพื้นฐานในภาษาคอมไพเตอร์บางภาษา	166
รูปที่ 7.4 บล็อกของคำสั่งที่มีการควบคุมลำดับการทำงานทั้งในกรณีปกติ และกรณีมีเหตุการณ์ยกเว้น	168
รูปที่ 7.5 โครงสร้างโปรแกรมในภาษา Pascal	169
รูปที่ 7.6 ขอบเขตและการมองเห็นของชื่อตัวแปร i	172
รูปที่ 7.7 แพ็กเกจส่วนอินเตอร์เฟซในภาษา Ada	173
รูปที่ 7.8 แพ็กเกจส่วนรายละเอียดคำสั่งในภาษา Ada	173
รูปที่ 7.9 ตัวอย่างการใช้แพ็กเกจ Date ในภาษา Ada	174
รูปที่ 7.10 เฮดเดอร์ไฟล์ date.h เพื่อคำนวณวันตามปฏิทิน	174
รูปที่ 7.11 ไฟล์ date.c เพื่อคำนวณวันตามปฏิทิน	175
รูปที่ 7.12 วิธีการใช้แพ็กเกจ date ในรูปแบบภาษา C	175
รูปที่ 8.1 การสร้างและการเรียกใช้โครงสร้างข้อมูลสแตกในภาษา C	186
รูปที่ 8.2 การสร้างและการเรียกใช้โครงสร้างข้อมูลสแตกในภาษา C++	187
รูปที่ 8.3 การใช้คลาสทำหน้าที่รวมข้อมูลเป็นหน่วยเดียว	189
รูปที่ 8.4 การเปลี่ยนโครงสร้างภายในของสแตกจากอาร์เรย์เป็นลิงค์ลิสต์	190
รูปที่ 8.5 ลำดับชั้นของการถ่ายทอดคลาส	190

รูปที่ 8.6 การถ่ายทอดคลาสในภาษา Java	191
รูปที่ 8.7 การประกาศคลาส Vector และ Stack ในภาษา Java	191
รูปที่ 8.8 การมีหลายรูปของฟังก์ชัน draw()	192
รูปที่ 9.1 สถานะของคอมพิวเตอร์ขณะประมวลผลโปรแกรม Modulo	205
รูปที่ 9.2 เปรียบเทียบการทำโปรแกรมเชิงคำสั่งและการทำโปรแกรมเชิงหน้าที่	206
รูปที่ 9.3 ฟังก์ชัน $f$ เชื่อมโยงค่า $x$ จากเซตของโดเมนไปยัง $f(x)$ ในเซตของเรนจ์	208
รูปที่ 9.4 รูปแบบต่างๆ ของการอธิบายฟังก์ชัน	209
รูปที่ 9.5 ความสัมพันธ์ที่เกิดขึ้นจากการใช้คำสั่ง (define x 5)	222
รูปที่ 9.6 โครงสร้างลิสต์ที่แสดงในลักษณะของแผนภูมิต้นไม้	223
รูปที่ 9.7 การนิยามฟังก์ชัน flip ที่เป็นฟังก์ชันอันดับสูง	228
รูปที่ 9.8 การคำนวณค่าดิฟเฟอเรนเชียล	231
รูปที่ 9.9 โปรแกรมคำนวณค่าสมการดิฟเฟอเรนเชียลในรูปแบบภาษา Scheme	231
รูปที่ 10.1 ตารางค่าความจริงของโอเปอเรเตอร์ที่ใช้ในนิพจน์พรอฟโพซิชัน	242
รูปที่ 10.2 นิพจน์และความหมายของนิพจน์ในตรรกศาสตร์เพรดิเคต	244
รูปที่ 10.3 คุณสมบัติของนิพจน์ในตรรกศาสตร์เพรดิเคต	245
รูปที่ 10.4 แผนภาพแสดงการพิสูจน์คำถามด้วยวิธีเรโซลูชัน	255
รูปที่ 10.5 ขั้นตอนการพิสูจน์ด้วยวิธีเรโซลูชันเมื่อคำถามเกี่ยวข้องกับกฎ	258
รูปที่ 10.6 ลำดับขั้นตอนการทำงานของโปรล็อกเพื่อพยายามพิสูจน์ความจริงของคำถาม <i>?-talkswith(bob, mary)</i>	259
รูปที่ 10.7 ขั้นตอนการทำงานของโปรล็อกเพื่อพิสูจน์ความจริงของคำถาม <i>?-talkswith(bob, allen)</i>	259
รูปที่ 10.8 แผนภาพแสดงโครงสร้างความสัมพันธ์ของบุคคลในครอบครัว	261
รูปที่ 10.9 กฎที่ใช้ในการคำนวณค่าดิฟเฟอเรนเชียล	269
รูปที่ 10.10 ขั้นตอนการทำงานของโปรแกรมคำนวณค่าดิฟเฟอเรนเชียล	270