	ใบความรู้ที่ 1	จำนวนชั่วโมง
	ชื่อวิชา โครงสร้างข้อมูลและอัลกอริทึม **รหัสวิชา** 30901-2001 (Data Structures and Algorithms)	5 ชั่วโมง
	ชื่อหน่วย หลักการของโครงสร้างข้อมูล (Principle of Data Structures)	

1.จุดประสงค์การเรียนรู้

1. อธิบายความหมายของโครงสร้างข้อมูลได้
2. บอกประเภทของโครงสร้างข้อมูลได้
3. อธิบายลักษณะโครงสร้างข้อมูลพื้นฐานแต่ละชนิดได้
4. เลือกใช้โครงสร้างข้อมูลพื้นฐานได้อย่างเหมาะสมและถูกต้อง
5. บอกวิธีการดำเนินการกับโครงสร้างข้อมูลได้

2. เนื้อหาสาระ

2.1. ความหมายของโครงสร้างข้อมูล

ข้อมูล (Data) คือ ข้อเท็จจริงต่าง ๆ ซึ่งอาจจะเป็นตัวเลขหรือไม่เป็นตัวเลขก็ได้

โครงสร้าง (Structure) คือ ความสัมพันธ์ของสมาชิกในกลุ่ม

โครงสร้างข้อมูล (Data Structure) เกิดจากคำสองคำ คือ “โครงสร้าง” และ “ข้อมูล” ซึ่งคำว่า “โครงสร้าง” เป็นความสัมพันธ์ระหวางสมาชิกในกลุ่ม ดังนั้นโครงสร้างข้อมูลจึงหมายถึงความสัมพันธ์ระหว่างข้อมูลที่ อยู่ในโครงสร้างนั้น สิ่งพื้นฐานในการประมวลผลข้อมูลคอมพิวเตอร์ คือ ข้อมูล (data) ดังนั้นการศึกษาถึงความสัมพันธ์ของข้อมูลจึงมีความสำคัญเป็นอย่างมากในศาสตร์ ทางคอมพิวเตอร์การศึกษาโครงสร้างข้อมูลประเภทต่าง ๆ จะต้องศึกษาสิ่งต่อไปนี้

1) นิยาม (Definition) เป็นการศึกษาความหมาย ความสัมพันธ์ระหวางข้อมูล และการดำเนินการในโครงสร้างข้อมูลประเภทนั้น

2) การนำไปใช้จริงในเครื่องคอมพิวเตอร์ (Implement)

2.2 ระดับของแบบชนิดข้อมูล

ระดับของแบบชนิดข้อมูล (Hierarchyof data types) หมายถึง ระดับของข้อมูลที่จำแนกตามแบบชนิดข้อมูลที่นำมาใช้กับเครื่องคอมพิวเตอร์ ซึ่งจำแนกข้อมูลออกเป็น 3ระดับ

2.2.1 ข้อมูลในระดับเครื่อง ข้อมูลในระดับเครื่อง (Hardwaredata types) เป็นระดับของแบบชนิดข้อมูลที่บันทึกในหน่วยความจำ ซึ่งข้อมูลจะถูกบันทึกด้วยรหัสแทนข้อมูลในภาษาเครื่อง เช่น

1) จำนวนเต็ม (Integer)

2) จำนวนจริง (real)

3) อักขระ (character)

2.2.2 ข้อมูลในระดับโปรแกรม ข้อมูลในระดับโปรแกรม (Virtual data types) เป็นระดับของแบบชนิดข้อมูลที่ดูเสมือนวามีจริง ' แต่ความ จริงเป็นการสร้างข้อมูลเทียมขึ้นมาใช้ โดยใช้ภาษาคอมพิวเตอร์สร้าง (built - in) ขึ้นมาเพื่อให้ความสะดวกแก่ผู้ใช้ ทำให้สามารถใช้งานได้คล่องตัวขึ้น ตัวอย่างข้อมูลในภาษา แบ่งออกเป็นชนิดต่าง ๆ ดังต่อไปนี้

1) ข้อมูลประเภทตัวเลขจำนวนเต็ม

2) ข้อมูลประเภทตัวเลขจำนวนจริง

3) ข้อมูลประเภท

4) ข้อมูลแบบผู้ใช้กำหนด (user–defined data types) เป็นแบบข้อมูลที่ผู้ใช้สามารถกำหนดเพิ่มขึ้นมาใช้เองได้ ซึ่งการกำหนดจะขึ้นอยู่กับภาษาที่

5) ข้อมูลแบบโครงสร้าง (Structure data types) เป็นข้อมูลที่สร้างขึ้นจากข้อมูลแบบสเกลาร์ หรืออาจจะสร้างขึ้นมาจากข้อมูลแบบโครงสร้างด้วย กันเองก็ ได้ (ยกเว้นไฟล์) ดังนี้

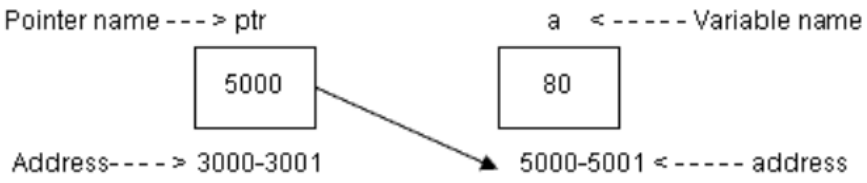
5.1) แถวลำดับ (array)

5.2) เซต (set)

5.3) ระเบียนข้อมูล (record)

5.4) แฟ้มข้อมูล (file)

6) ข้อมูลแบบพ้อยน์เตอร์(Pointer data types) ข้อมูลพ้อยน์เตอร์ คือ ตัวแปรชนิดหนึ่งที่เก็บตำแหน่ง (Address) ของข้อมูลภายใน หน่วยความจำ ซึ่งการเก็บตำแหน่ง จะเก็บเฉพาะตำแหน่งแรกของข้อมูลเท่านั้น



2.2.3 ข้อมูลในระดับความคิด (Abstract data types) เป็นระดับของแบบชนิดข้อมูลประเภทนามธรรมที่สร้างขึ้น จากจินตนาการของผู้ใช้ เป็นแบบชนิดข้อมูลที่ไม่มีรูปร่างหรือลักษณะให้เห็น การอธิบายลักษณะของข้อมูลจะใช้

สัญลักษณ์ ถ้าต้องการทำให้แบบชนิดข้อมูลที่เป็นนามธรรมเป็นข้อมูลที่เป็นรูปธรรม ก็ต้องนำไปใช้จริงกับเครื่องคอมพิวเตอร์จริง (Implement) ข้อมูลในระดับความคิดแบ่งออกเป็น 2 ประเภท คือ

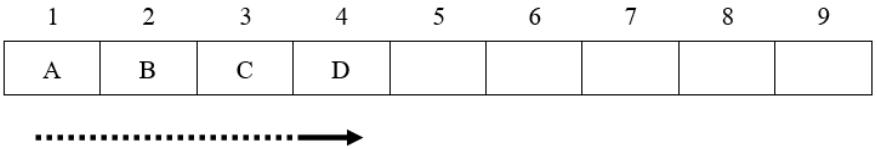
(1) โครงสร้างข้อมูลแบบเชิงเส้น (Linear data structures) เช่น

1) ลิสต์ (list)

2) สแตก (stack)

3) คิว (queue)

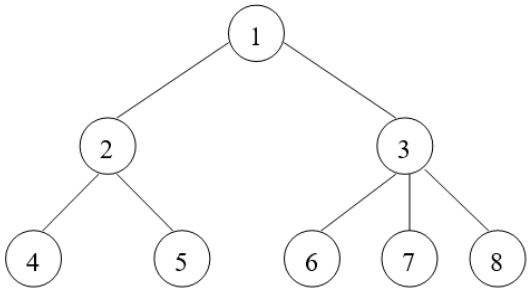
4) ดีคิว (deque)



(2) โครงสร้างข้อมูลแบบไม่ใช่เชิงเส้น (Non-lineardata structures) เช่น

1) ทรี (tree)

2) กราฟ (graph)



จากข้อมูลในแต่ละระดับที่กล่าวมา ข้อมูลในระดับที่ 2 (ข้อมูลในระดับความคิด) นักเขียนโปรแกรม (Programmer) เป็นผู้เขียนโปรแกรมสร้างข้อมูลขึ้นมาเพื่อใช้เป็นตัวเชื่อมความสัมพันธ์ไปสู่ข้อมูลในระดับที่ 1 (ข้อมูลใน ระดับโปรแกรม) ส่วนโปรแกรมแปลภาษาทำหน้าที่เป็นตัวเชื่อมความสัมพันธ์จากข้อมูลในระดับที่1 ไปสู่ข้อมูลในระดับ ที่ 0 (ข้อมูลในระดับเครื่อง) ซึ่งเป็นระดับที่เครื่องบันทึกข้อมูลโดยใช้รหัสแทนข้อมูลจริง

2.2.4.การแทนที่ข้อมูลในหน่วยความจำหลัก การประมวลผลข้อมูลด้วยคอมพิวเตอร์ ข้อมูลที่ต้องการประมวลผลจะถูกนำไปเก็บในหน่วยความจำหลักเพื่อ ประมวลผล ในภาษาคอมพิวเตอร์ระดับสูงจะต้องมีวิธีการจัดการกับหน่วยความจำหลัก เพื่อนำหน่วยความจำหลักไปใช้ ในโครงสร้างข้อมูลนั้น และเมื่อไม่มีการใช้เนื้อที่ในหน่วยความจำหลักนั้นแล้วควรจะต้องมีการคืนเนื้อที่ใน หน่วยความจำหลักด้วย เพื่อนำเนื้อที่หน่วยความจำหลักที่

ไม่ได้ใช้สามารถนำกลับมาใช้ใหม่ได้ โดยทั่วไปการเขียน โปรแกรมคอมพิวเตอร์มีการแทนที่ข้อมูลในหน่วยความจำหลักอยู่ 2 วิธี คือ

1) การแทนที่ข้อมูลแบบสแตติก การแทนที่ข้อมูลแบบสแตติก (Static memory representation) เป็นการแทนที่ข้อมูลที่มีการจองเนื้อที่ แบบคงที่แน่นอน การแทนที่แบบนี้ต้องมีการกำหนดขนาดก่อนการใช้งาน ข้อเสียของการแทนที่ด้วยวิธีนี้ก็คือไม่ สามารถปรับขนาดให้เพิ่มขึ้นหรือลดลงได้ ไม่สามารถเกบข้อมูลเก็บขนาดเนื้อที่ที่กำหนดไว้ และถ้ากำหนดขนาดเนื้อที่ ไว้มากเกินจำเป็นทั้ง ๆ ที่มีข้อมูลอยู่จำนวนน้อยจะทำให้สูญเสียเนื้อที่โดยเปล่าประโยชน์ โครงสร้างข้อมูลที่มีการแทนที่ ในหน่วยความจำหลักด้วยวิธีนี้คือ แถวลำดับ

2) การแทนที่ข้อมูลแบบไดนามิก การแทนที่ข้อมูลแบบไดนามิก (dynamic memory representation) เป็นการแทนที่ข้อมูลที่ไม่ต้องจองเนื้อที่ และขนาดของเนื้อที่ที่นำมาใช้สามารถยืดหยุ่นได้ ตามความต้องการของผู้ใช้ คือ ถ้าข้อมูลมีน้อยก็ใช้เนื้อที่น้อย และถ้าข้อมูลมีมากกสามารถใช้เนื้อที่มากตามที่ใช้จริงได้ นอกจากนั้นส่วนเนื้อที่ในหน่วยความจำหลักที่ไม่ใช้แล้ว สามารถส่งคืนเพื่อกลับมาใช้ใหม่ได้อีก ภาษาคอมพิวเตอร์ระดับสูงบางภาษาเท่านั้นที่สามารถแทนที่ข้อมูลด้วยวิธีนี้ เช่น ภาษาปาสคาล ภาษาซี ภาษาพีแอลวัน และภาษาอัลกอล เป็นต้น สำหรับโครงสร้างข้อมูลที่มีการแทนที่ใน หน่วยความจำหลักแบบ ไดนามิก คือ ตัวชี้ หรือ พอยน์เตอร์ (pointer)

2.3โครงสร้างข้อมูลพื้นฐาน

โครงสร้างข้อมูลพื้นฐานบางครั้งเรียกว่า ชนิดข้อมูลพื้นฐาน เป็นโครงสร้างข้อมูลที่มีค่าเฉพาะ ไม่มีโครงสร้างข้อมูลอื่นมาเป็นส่วนประกอบ มีอยู่ในภาษาคอมพิวเตอร์ทั่วไป ผู้ใช้ไม่ต้องสร้างขึ้นใหม่เมื่อต้องการเก็บค่าสามารถเรียกใช้งานได้ทันทีได้แก่

2.3.1. ข้อมูลชนิดเลขจำนวนเต็ม

ข้อมูลชนิดเลขจำนวนเต็ม(Integer) หมายถึง จำนวนเต็มบวก จำนวนเต็มศูนย์ และจำนวนเต็มลบ เป็นข้อมูลที่สามารถนำไปใช้ในการคำนวณได้ แบ่งออกเป็นชนิดย่อยๆ ได้ดังนี้

การคำนวณทางคณิตศาสตร์ของข้อมูลชนิดเลขจำนวนเต็ม ใช้เครื่องหมาย + - * / mod และ div ถ้าตัวแปรที่จะทำการคำนวณเป็นเลขจำนวนเต็ม ผลลัพธ์ที่ได้จะเป็นดังนี้

จำนวนเต็ม	+	จำนวนเต็ม	ผลลัพธ์ที่ได้เป็น	จำนวนเต็ม
จำนวนเต็ม	-	จำนวนเต็ม	ผลลัพธ์ที่ได้เป็น	จำนวนเต็ม
จำนวนเต็ม	*	จำนวนเต็ม	ผลลัพธ์ที่ได้เป็น	จำนวนเต็ม
จำนวนเต็ม	/	จำนวนเต็ม	ผลลัพธ์ที่ได้เป็น	จำนวนจริง
จำนวนเต็ม	mod	จำนวนเต็ม	ผลลัพธ์ที่ได้เป็น	จำนวนเต็ม

จำนวนเต็ม div จำนวนเต็ม ผลลัพธ์ที่ได้เป็น จำนวนเต็ม

2.3.2. ข้อมูลชนิดเลขจำนวนจริง

ในระบบคอมพิวเตอร์ข้อมูลชนิดเลขจำนวนจริง (Real หรือ Floating Point) หมายถึง ตัวเลขที่มีทศนิยม อาจมีเครื่องหมายบวก หรือเครื่องหมายลบ สามารถนำไปใช้ในการคำนวณได้ และสามารถใช้แทนตัวเลขที่มีขนาดใหญ่มากๆ หรือเล็กมากๆ ได้ ในขณะที่เลขจำนวนเต็มไม่สามารถทำได้ การเขียนตัวเลขทศนิยมนี้จะเขียนอยู่ในรูป e ยกกำลัง เช่น 30.5 จะแสดงออกมาเป็น 3.05E+1

การคำนวณทางคณิตศาสตร์ของข้อมูลชนิดเลขจำนวนจริง จะใช้เครื่องหมาย + - * และ / ถ้าตัวแปรที่ใช้ในการคำนวณเป็นเลขจำนวนจริงและจำนวนเต็ม ผลลัพธ์ที่ได้จะเป็นดังนี้

จำนวนจริง	+	จำนวนจริง	ผลลัพธ์ที่ได้เป็น	จำนวนจริง
จำนวนจริง	-	จำนวนจริง	ผลลัพธ์ที่ได้เป็น	จำนวนจริง
จำนวนจริง	*	จำนวนจริง	ผลลัพธ์ที่ได้เป็น	จำนวนจริง
จำนวนจริง	/	จำนวนจริง	ผลลัพธ์ที่ได้เป็น	จำนวนจริง
จำนวนจริง	+	จำนวนเต็ม	ผลลัพธ์ที่ได้เป็น	จำนวนจริง
จำนวนจริง	-	จำนวนเต็ม	ผลลัพธ์ที่ได้เป็น	จำนวนจริง
จำนวนจริง	*	จำนวนเต็ม	ผลลัพธ์ที่ได้เป็น	จำนวนจริง
จำนวนจริง	/	จำนวนเต็ม	ผลลัพธ์ที่ได้เป็น	จำนวนจริง

สรุปได้ว่า การคำนวณทางคณิตศาสตร์ของเลขจำนวนจริงผลลัพธ์ที่ได้จะเป็นเลขจำนวนจริง

ลำดับการคำนวณทางคณิตศาสตร์

การคำนวณหาค่านิพจน์ทางคณิตศาสตร์ จะทำการคำนวณตามลำดับความสำคัญของเครื่องหมายจากลำดับความสำคัญสูงสุดไปสู่ความสำคัญต่ำสุด ดังนี้

1. กระทำการภายในวงเล็บ ถ้ามีวงเล็บซ้อนกันหลายๆ วงเล็บให้ทำวงเล็บที่อยู่ด้านในสุดออกมาด้านนอก
2. ทำเลขยกกำลัง
3. ทำเครื่องหมาย *, /, mod, div
4. ทำเครื่องหมาย +, -

กรณีที่นิพจน์ประกอบไปด้วยเครื่องหมายที่มีลำดับความสำคัญเท่ากัน ลำดับของการกระทำจะกระทำเครื่องหมายที่พบก่อนจากซ้ายไปขวา ยกเว้น เลขยกกำลังให้ทำจากขวามาซ้าย

2.3.3. ข้อมูลชนิดตัวอักขระ

ข้อมูลชนิดตัวอักขระ (Character) แบ่งออกเป็น 2 ประเภท คือ

3.1 ตัวอักขระ (Character) หมายถึง ตัวอักษรหรือสัญลักษณ์พิเศษต่างๆ ที่มีความยาว 1 ตัวอักษรเท่านั้น โดยจะมีขนาด 1 ไบต์ (8 บิต) ต่อ 1 ตัวอักษร ข้อมูลตัวอักขระจะต้องเขียนอยู่ในเครื่องหมาย ‘ ’ (single quote) เช่น ประกาศตัวแปรชนิดอักขระด้วยภาษาซี ดังนี้

```
char   A, B   /*  ประกาศตัวแปร  A และ B  เป็นชนิดตัวอักขระเก็บข้อมูล 1 ตัว  */
```

การกำหนดค่าให้กับตัวแปรทำได้ดังนี้

```
A  =  '*';          /*  กำหนดให้ตัวแปร  A  เก็บสัญลักษณ์  *  1  ตัว  */
B  =  'Y';          /*  กำหนดให้ตัวแปร  B  เก็บตัวอักษร  Y  1  ตัว  */
```

3.2 ข้อความ (String) หมายถึง ตัวอักขระที่มีความยาวมากกว่า 1 ตัว เรียงติดต่อกันเป็นข้อความ การเขียนข้อมูลข้อความจะต้องเขียนอยู่ในเครื่องหมาย “ ” (double quote) เช่น

```
char  fname[30];       /*  ประกาศตัวแปร fname เป็นชนิดข้อความ
                                ขนาด 30  ตัวอักษร  */
char  title[15];       /*  ประกาศตัวแปร  title  เป็นชนิดข้อความ
                                ขนาด 10  ตัวอักษร  */
```

การกำหนดค่าให้กับตัวแปรทำได้ดังนี้

```
fname  = "SUTREE";  /* กำหนดให้ตัวแปร fname เก็บคำว่า  SUTREE   */
title   = "Report";  /*  กำหนดให้ตัวแปร title เก็บคำว่า  Report   */
```

2.3.4. ข้อมูลชนิดบูลีน

ข้อมูลชนิดบูลีน (Boolean) เป็นข้อมูลที่ใช้ในเชิงการตัดสินใจ โดยมีการกำหนดเงื่อนไขในการตรวจสอบค่าผลลัพธ์ที่ได้มีค่าเพียง 2 ค่า คือ True (จริง) หรือ False (เท็จ) เท่านั้น
ตัวแปรชนิดนี้จะมีประโยชน์ต่อการออกแบบโปรแกรมและควบคุมการทำงานในโปรแกรม

2.4 การดำเนินการกับโครงสร้างข้อมูล

การดำเนินการกับโครงสร้างข้อมูล (Data Structure Operation) จะขึ้นอยู่กับลักษณะของการใช้งาน ซึ่งวิธีการที่ใช้กันมากมีด้วยกัน 4 ประการ คือ

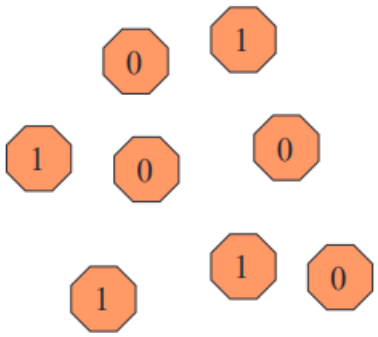
1. **การเข้าถึงแฟ้มข้อมูล (Traversing or Access)** เป็นการเข้าไปติดต่อกับระเบียน ข้อมูล (Record) ที่ต้องการ เพื่อทำการเปลี่ยนแปลงหรือแก้ไขข้อมูลในระเบียนนั้น
2. **การค้นหา (Searching)** เป็นการค้นหาระเบียนข้อมูลตามค่าที่กำหนด เพื่อทำการเรียกใช้หรือเปลี่ยนแปลงแก้ไขข้อมูลตามเงื่อนไข
3. **การเพิ่ม (Inserting)** เป็นการเพิ่มระเบียนข้อมูลใหม่เข้าไปในแฟ้มข้อมูล
4. **การลบ (Deleting)** เป็นการลบระเบียนข้อมูลที่ไม่ต้องการออกจากแฟ้มข้อมูล เพื่อปรับปรุงข้อมูลให้ทันสมัย

Data Structure •

โครงสร้างข้อมูล หมายถึง • เป็นการรวมกันของแต่ละหน่วยย่อย ซึ่งแต่ละหน่วยย่อยคือ ชนิด ข้อมูล (data type) หรือโครงสร้างข้อมูลอื่นๆ • เซตของการมีส่วนร่วมหรือมีความสัมพันธ์กัน ก่อให้เกิดการรวมกัน ของหน่วยย่อย

ตัวอย่าง

ข้อมูล+ ไม่มีโครงสร้าง กลุ่มของ Bit ข้อมูลในระบบ Binary แบบไม่มีโครงสร้าง

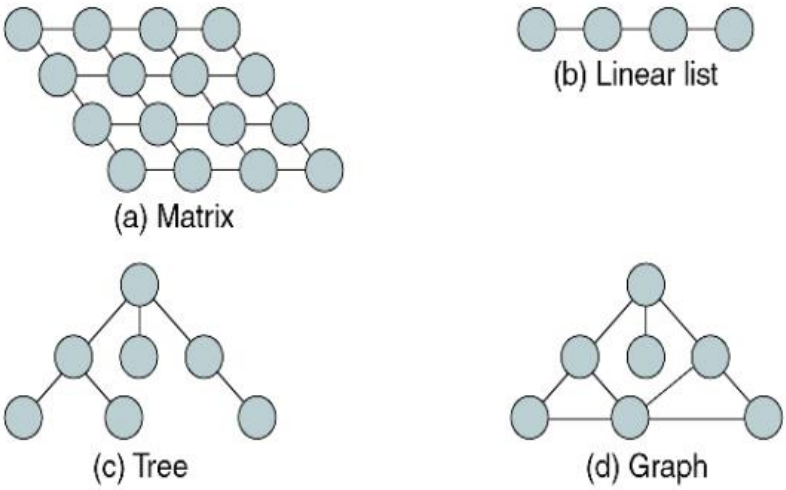


แสดงข้อมูล ข้อมูล+ โครงสร้าง กลุ่มของ Bit ข้อมูล1 Byte ในระบบ Binary ที่นำมาเรียงกันแล้วทำให้มีความหมาย ในตัวอย่างนี้หมายถึง เลข65 หากใน ระบบ ASCII จะหมายถึงตัวอักษร ‘A’

มีค่า = 65

Array	Record
Homogeneous sequence of data or data types known as elements	Heterogeneous combination of data into a single structure with an identified key
Position association among the elements	No association

ที่มา: http://202.29.14.206/staff/keng/4132303/chapter1.pdf

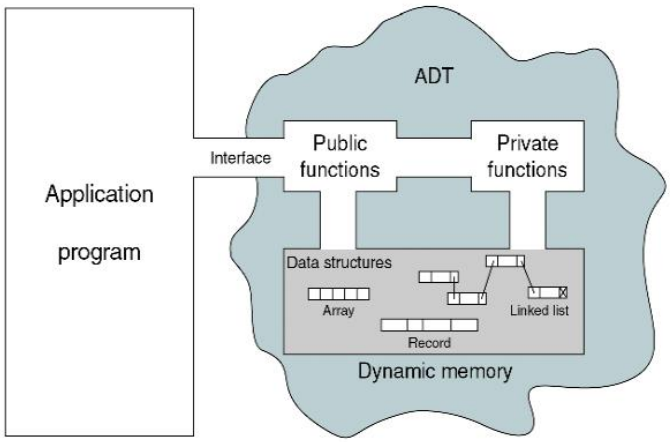


ที่มา: http://202.29.14.206/staff/keng/4132303/chapter1.pdf

Abstract Data Type (ADT)

- ***ชนิดข้อมูลนามธรรม (ADT)*** : เป็นการกำหนดคุณลักษณะเฉพาะ เซตข้อมูล (set of data) และเซตของการดำเนินการ (set of operations) ที่สามารถกระทำกับข้อมูลได้
  - Declaration of data
  - Declaration of operations
  - Encapsulation of data and operations
- อาจกล่าวได้ว่าการโปรแกรมเชิงวัตถุ (Object-Oriented Programming) เป็นเรื่องเดียวกับนามธรรม โดยมีคำหลักสองคำคือ ข้อมูล (data) และ กระบวนการในการจัดการกับข้อมูล (method or operation)

Model for an ADT



http://202.29.14.206/staff/keng/4132303/chapter1.pdf

โครงสร้างข้อมูลแบบอาร์เรย์และสตริง

โครงสร้างข้อมูลแบบอาร์เรย์ เป็นโครงสร้างข้อมูลประเภทเชิงเส้นที่มีรูปแบบการจัดเก็บข้อมูลแบบเรียงลำดับต่อเนื่องกัน โดยอาร์เรย์จะมีการเก็บข้อมูลในลักษณะของตารางเป็นช่องๆ แต่ละช่องมีขนาดเท่ากันและเก็บข้อมูลชนิดเดียวกัน

กล่าวโดยสรุป โครงสร้างข้อมูลแบบอาร์เรย์ คือ โครงสร้างข้อมูลที่ภายในตัวแปรเดียวสามารถเก็บข้อมูลได้มากกว่า 1 ค่า แต่ละค่ามีขนาดเท่ากันและเป็นข้อมูลชนิดเดียวกัน โดยเราสามารถเข้าถึงค่าต่าง ๆ ในอาร์เรย์ด้วยการใช้เลขดัชนี (Index) หรือซับสคริปต์ (Subscript) เป็นตัวอ้างอิงตำแหน่งของสมาชิกในอาร์เรย์

ลักษณะของอาร์เรย์

อาร์เรย์เป็นโครงสร้างที่มีลักษณะสำคัญดังนี้

1. เป็นโครงสร้างเชิงเส้น คือ มีรูปแบบการจัดเก็บข้อมูลเป็นแนวเส้นตรงที่ต่อเนื่องกัน
2. ข้อมูลแต่ละช่องในอาร์เรย์จะต้องเป็นชนิดเดียวกัน เช่น กำหนดให้อาร์เรย์ T เก็บข้อมูลชนิดตัวเลขจำนวนเต็ม ดังนั้นข้อมูลทุกช่องในอาร์เรย์ T จะเป็นตัวเลขจำนวนเต็มเท่านั้น จะเป็นตัวอักขระหรือตัวเลขจำนวนจริงไม่ได้
3. อาร์เรย์เป็นโครงสร้างที่มีขนาดคงที่ (Static Structure) กล่าวคือ เมื่ออาร์เรย์ N ถูกสร้างขึ้นให้เป็นอาร์เรย์ที่มีขนาด 50 ช่อง นั่นก็หมายความว่า อาร์เรย์ N จะมีหน่วยความจำ 50 ช่องเรียงติดต่อกันเพื่อจัดเก็บข้อมูล ซึ่งจะเก็บมากกว่า 50 ช่องไม่ได้ ดังนั้นโครงสร้างข้อมูลแบบอาร์เรย์จึงต้องจองเนื้อที่ในหน่วยความจำให้เพียงพอต่อจำนวนสมาชิก
4. การอ้างอิงสมาชิกในอาร์เรย์สามารถเข้าถึงได้โดยตรง (Direct Accress) เช่น ต้องการเข้าถึงข้อมูลในช่องที่ 5 ของอาร์เรย์ N ซึ่งมีทั้งหมด 10 ช่อง ก็เพียงแต่กำหนด N[5] เท่านั้นก็สามารถเข้าถึงข้อมูลที่อยู่ในช่องที่ 5 ได้ทันที โดยไม่ต้องผ่านข้อมูลก่อนหน้า

การอ้างอิงตำแหน่งสมาชิกในอาร์เรย์

การอ้างอิงตำแหน่งสมาชิกในอาร์เรย์จะต้องเริ่มต้นด้วยชื่ออาร์เรย์และตามด้วยเลขดัชนี โดยเลขดัชนีจะอยู่ภายในเครื่องหมาย [ ] หรือ ( )

ตัวอย่างเช่น กำหนดอาร์เรย์ชื่อ Name มีขนาด 40 ช่องเพื่อเก็บชื่อนักศึกษา

Name[1] หมายถึง ชื่อนักศึกษาคนที่ 1

Name[2] หมายถึง ชื่อนักศึกษาคนที่ 2

:

Name[40] หมายถึง ชื่อนักศึกษาคนที่ 40

อย่างไรก็ตามในภาษาคอมพิวเตอร์อย่างเช่นภาษา C หรือ JAVA หมายเลขลำดับของอาร์เรย์จะเริ่มต้นด้วยหมายเลข 0 ดังนั้นหากมีการประกาศตัวแปรอาร์เรย์ด้วยภาษา C หรือ JAVA ในการอ้างอิงลำดับสมาชิกในอาร์เรย์อาจเกิดความสับสนได้จึงต้องใช้อย่างระมัดระวัง

ตัวอย่าง ประกาศตัวแปรอาร์เรย์ชื่อ A ในภาษา C ดังนี้

```
int A[5];
```

จะมีการจองพื้นที่ในหน่วยความจำสำหรับตัวแปรอาร์เรย์ชื่อ A จำนวน 5 ช่อง คือ A[0], A[1], A[2], A[3], A[4] ซึ่งแสดงในรูปของตารางได้ดังนี้

เลขดัชนี	0	1	2	3	4
การอ้างอิง	A[0]	A[1]	A[2]	A[3]	A[4]

แสดงรายละเอียดของอาร์เรย์ A

จากตัวอย่าง อาร์เรย์ A ขนาด 5 ช่อง เลขดัชนีจะเรียงตามลำดับโดยเริ่มจาก 0 ถึง 4 ในการอ้างอิงตำแหน่งเพื่อจัดเก็บข้อมูลในอาร์เรย์ทำได้ตัวอย่างต่อไปดังนี้

```
A[0] = 5;    /* ให้ใส่ค่า 5 ลงในอาร์เรย์ A ในตำแหน่ง 0 (คือช่องที่ 1 ในอาร์เรย์)
A[1] = 10;   /* ให้ใส่ค่า 10 ลงในอาร์เรย์ A ในตำแหน่ง 1 (คือช่องที่ 2 ในอาร์เรย์)
A[2] = 15;   /* ให้ใส่ค่า 15 ลงในอาร์เรย์ A ในตำแหน่ง 2 (คือช่องที่ 3 ในอาร์เรย์)
A[3] = 20;   /* ให้ใส่ค่า 20 ลงในอาร์เรย์ A ในตำแหน่ง 3 (คือช่องที่ 4 ในอาร์เรย์)
A[4] = 25;   /* ให้ใส่ค่า 25 ลงในอาร์เรย์ A ในตำแหน่ง 4 (คือช่องที่ 5 ในอาร์เรย์)
```

เลขดัชนี	0	1	2	3	4
ค่าข้อมูล	5	10	15	20	25
การอ้างอิง	A[0]	A[1]	A[2]	A[3]	A[4]

แสดงการจัดเก็บข้อมูลในอาร์เรย์ A

แสดงการจัดเก็บข้อมูลในอาร์เรย์ A

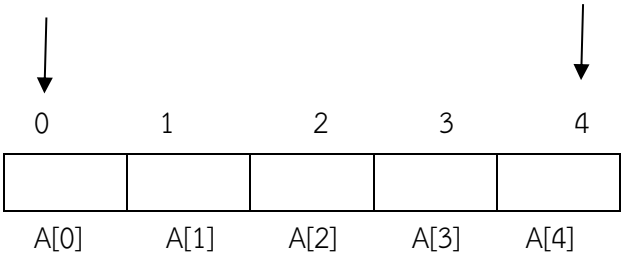
ขอบเขตของอาร์เรย์ (Bounds)

การกำหนดตัวแปรอาร์เรย์ เลขดัชนีในอาร์เรย์ประกอบด้วยช่วงขอบเขตของค่า ซึ่งประกอบด้วยขอบเขตล่างสุด (Lower Bound) ใช้ระบุตำแหน่งเริ่มต้นของการจัดเก็บข้อมูลและขอบเขตบนสุด (Upper Bound) ใช้กำหนดตำแหน่งสูงสุดในการเก็บค่าข้อมูลของอาร์เรย์ แต่อย่างไรก็ตามในภาษาคอมพิวเตอร์บางภาษาจะกำหนดได้เพียงขอบเขตบนสุดเท่านั้น เนื่องจากขอบเขตล่างสุดได้ถูกกำหนดคงที่ไว้อยู่แล้ว เช่น ภาษา C และ JAVA จะถูกกำหนดของเขตล่างสุดเท่ากับ 0 ในขณะที่ภาษา FORTRAN จะถูกกำหนดขอบเขตล่างสุดเท่ากับ 1 ส่วนภาษา PASCAL หรือ PL/1 สามารถกำหนดขอบเขตล่างสุดและขอบเขตบนสุดของอาร์เรย์ได้เอง และยังสามารถกำหนดให้มีค่าติดลบได้อีกด้วย

ตัวอย่าง การกำหนดตัวแปรอาร์เรย์ในภาษา C

```
int  A[5];
```

Lower bound = 0 Upper bound = 4

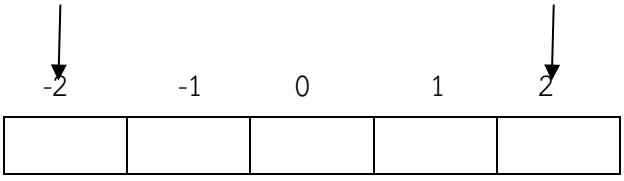


รูปแบบรายละเอียดของอาร์เรย์ A

ตัวอย่าง การกำหนดตัวแปรอาร์เรย์ในภาษา PASCAL

```
VAR  N = Array[-2..2]  of  Integer;
```

Lower bound = -2 Upper bound = 2



รูปแบบรายละเอียดของอาร์เรย์ N

ชนิดของอาร์เรย์

การกำหนดชนิดของอาร์เรย์ตามรูปแบบโครงสร้าง มีอยู่ด้วยกัน 3 รูปแบบ ได้แก่

1. อาร์เรย์ 1 มิติ (One Dimension Array)
2. อาร์เรย์ 2 มิติ (Two Dimension Array)
3. อาร์เรย์ 3 มิติ (Three Dimension Array)

อาร์เรย์ 1 มิติ (One Dimension Array)

จะมีการจัดเก็บข้อมูลในลักษณะต่อเนื่องกันเป็นแถว การอ้างอิงหรือการเข้าถึงสมาชิกตัวใดตัวหนึ่งจะใช้เลขดัชนี (Index) เพียง 1 ตัวเท่านั้น การแสดงอาร์เรย์ 1 มิติในรูปแบบช่องตารางจะมีลักษณะเป็นตารางแถวเดียว ซึ่งจะจัดแนวของช่องแบบแนวนอนหรือแนวตั้งก็ได้

1	A[1]
2	A[2]
3	A[3]
4	A[4]

แสดงโครงสร้างอาร์เรย์แนวตั้ง

1	2	3	4
A[1]	A[2]	A[3]	A[4]

แสดงโครงสร้างอาร์เรย์แนวนอน

รูปแบบทั่วไปของอาร์เรย์ 1 มิติ

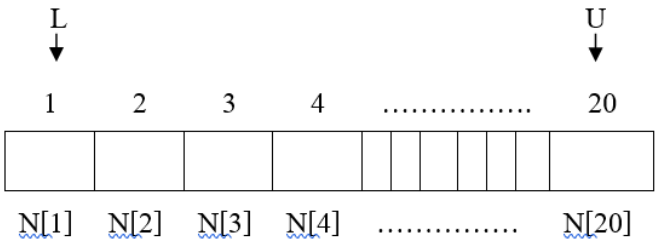
```
A [L : U]
```

โดยที่ A คือ ชื่อของอาร์เรย์

L คือ ขอบเขตล่างสุด (Lower Bound)

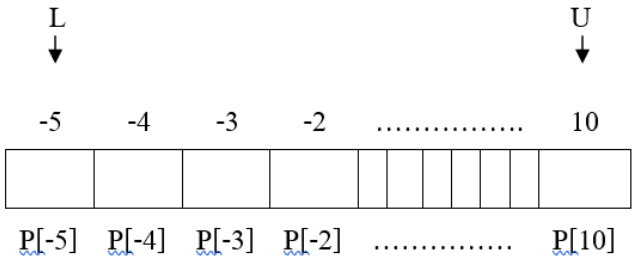
U คือ ขอบเขตบนสุด (Upper Bound)

ตัวอย่าง 1 อาร์เรย์ N[1 : 20] มีความหมายว่า อาร์เรย์ ชื่อ N , L = 1, U = 20



รูปแบบรายละเอียดของอาร์เรย์ N

ตัวอย่าง 2 อาร์เรย์ P[-5 : 10] มีความหมายว่า อาร์เรย์ ชื่อ P L = -5, U = 10



การอ้างอิงสมาชิกและการกำหนดค่าข้อมูลในอาร์เรย์ 1 มิติ

การอ้างอิงสมาชิกในอาร์เรย์ 1 มิติ จะใช้รูปแบบดังนี้คือ

A[i]

โดยที่ A คือ ชื่ออาร์เรย์ i คือ เลขดัชนี ใช้ระบุตำแหน่งใดๆ

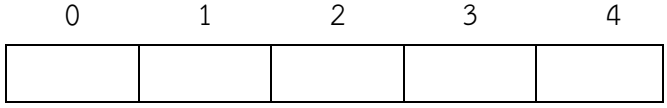
เพื่อให้เห็นการอ้างอิงถึงสมาชิกและการกำหนดค่าข้อมูลให้กับสมาชิกในอาร์เรย์ 1 มิติ ให้พิจารณาตัวอย่างต่อไปนี้

ตัวอย่าง กำหนดอาร์เรย์ score[1 : 5] หมายถึง อาร์เรย์ score ที่มีจำนวนช่องตั้งแต่ 1 ถึง 5 ซึ่งเราสามารถเขียนอีกรูปแบบหนึ่งว่า score[5] ก็ได้ และถ้าหากใช้ภาษา C กำหนดให้ score คือ อาร์เรย์ 1 มิติเก็บข้อมูลชนิดตัวเลขจำนวนเต็ม จะกำหนดได้ดังนี้

```
int score[5];
```

การกำหนดดังกล่าวจะมีการจองพื้นที่ในหน่วยความจำขนาด 5 ช่อง โดยเริ่มจาก

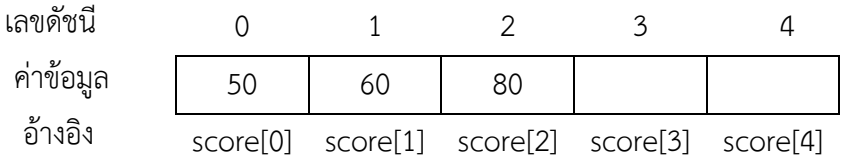
Score[0], score[1], score[2], score[3], score[4] แสดงเป็นตารางได้ดังรูป



score[0] score[1] score[2] score[3] score[4]

การอ้างอิงสมาชิกและการกำหนดค่าข้อมูลในอาร์เรย์ score ทำได้ดังนี้

```
Score[0] = 50;    /* ให้ใส่ค่า 50 ลงในอาร์เรย์ score ในตำแหน่ง 0 (ช่องที่ 1)
Score[1] = 60;    /* ให้ใส่ค่า 60 ลงในอาร์เรย์ score ในตำแหน่ง 1 (ช่องที่ 2)
Score[2] = 80;    /* ให้ใส่ค่า 80 ลงในอาร์เรย์ score ในตำแหน่ง 2 (ช่องที่ 3)
X = score[2];     /* ให้นำข้อมูลในอาร์เรย์ score ในตำแหน่งที่ 2 มาไว้ในตัวแปร X
```



แสดงรายละเอียดการจัดเก็บข้อมูลในอาร์เรย์ score

ตัวอย่าง ต้องการเก็บส่วนสูงของนางงามหมายเลข 115 ถึง 124 ดังนี้ 170, 169, 175, 171, 174, 170, 168, 166, 172, 173

จากความต้องการสามารถเก็บข้อมูลในรูปแบบของอาร์เรย์ 1 มิติได้ โดยกำหนดชื่ออาร์เรย์เป็น H และกำหนดขอบเขตล่างสุดเริ่มต้นที่ 115 ขอบเขตบนสุดเป็น 124 กำหนดรูปแบบของอาร์เรย์ตามโครงสร้าง ได้ดังนี้

H[115 : 124]

และมีการจัดเก็บข้อมูลในตำแหน่งต่างๆ ของอาร์เรย์ ดังนี้

H[115] = 170 H[116] = 169 H[117] = 175 H[118] = 171 H[119] = 174

H[120] = 170 H[121] = 168 H[122] = 166 H[123] = 172 H[124] = 173

115	116	117	118	119	120	121	122	123	124
170	169	175	171	174	170	168	166	172	173
H[115]	H[116]	H[117]	H[118]	H[119]	H[120]	H[121]	H[122]	H[123]	H[124]

แสดงรายละเอียดการจัดเก็บข้อมูลในอาร์เรย์ในรูปแบบตาราง

จากตัวอย่างจะเห็นได้ว่าเลขดัชนีที่ใช้ระบุตำแหน่งที่ใช้ในการจัดเก็บข้อมูลในอาร์เรย์ไม่จำเป็นต้องเริ่มจาก 1 เสมอไปอาจเริ่มเก็บจากตำแหน่งใดๆ ก็ได้

การคำนวณในอาร์เรย์ 1 มิติ

การคำนวณในอาร์เรย์ 1 มิติ จะมีการคำนวณเพื่อหาค่าอยู่ 3 กรณี ดังนี้

1. การคำนวณหาจำนวนช่องของอาร์เรย์หรือจำนวนสมาชิกของอาร์เรย์
2. การคำนวณหาขนาดของอาร์เรย์
3. การคำนวณหาตำแหน่งแอดเดรส (Address) ของ A[i]

การคำนวณหาจำนวนช่องของอาร์เรย์หรือจำนวนสมาชิกของอาร์เรย์

สามารถคำนวณได้จากสูตรดังนี้

จำนวนช่องของ A[L:U] = U – L + 1

ตัวอย่าง จงคำนวณหาจำนวนช่องของอาร์เรย์ M[-2 : 5]

จำนวนช่อง A[L : U] = U – L + 1

M[-2 : 5] = 5 – (-2) + 1

= 8

นั่นคือ อาร์เรย์ M[-2 : 5] มีจำนวนช่องทั้งหมด 8 ช่อง

การคำนวณหาขนาดของอาร์เรย์

การหาขนาดของอาร์เรย์ เป็นการคำนวณเพื่อให้ทราบว่าจะใช้พื้นที่หน่วยความจำเท่าไรในการจัดเก็บข้อมูลทั้งหมดของอาร์เรย์ ซึ่งคำนวณได้จากสูตร ดังนี้

ขนาดของอาร์เรย์ = จำนวนช่องของอาร์เรย์ x ขนาดความกว้างของแต่ละช่อง

จากตัวอย่างข้างต้นถ้าแต่ละช่องของอาร์เรย์ M มีขนาดความกว้าง 2 Byte อาร์เรย์ M จะใช้พื้นที่ในหน่วยความจำเพื่อจัดเก็บข้อมูลทั้งหมด 16 Byte (8 x 2)

ตัวอย่าง อาร์เรย์ A[5 : 19] แต่ละช่องมีขนาดความกว้าง 4 Byte จงหาขนาดของอาร์เรย์ A

วิธีทำ 1. หาจำนวนช่องของอาร์เรย์ A[L:U] = U – L + 1

A[5:19] = 19 – 5 + 1

= 15 ช่อง

2. หาขนาดของอาร์เรย์ = จำนวนช่อง x ขนาดความกว้างแต่ละช่อง

= 15 x 4

= 60 Byte

นั่นคือ อาร์เรย์ A[5 : 19] มีขนาด 60 Byte สรุปได้ว่าอาร์เรย์ A[5 : 19] ใช้เนื้อที่ในหน่วยความจำไปทั้งสิ้น 60 Byte

การคำนวณหาตำแหน่งแอดเดรส (Address) ของ A[i]

เป็นการหาแอดเดรสในตำแหน่ง A[i] ของอาร์เรย์ A ที่จัดเก็บในหน่วยความจำของคอมพิวเตอร์ ทำได้โดยเอาขนาดความกว้างของแต่ละช่องของอาร์เรย์คูณจำนวนช่องก่อนถึง A[i] บวกด้วยตำแหน่งแอดเดรสเริ่มต้นของอาร์เรย์ ก็จะได้ตำแหน่งของแอดเดรส A[i]

A[1]	A[2]	A[3]	.....	A[i]	.....	A[n]

พื้นที่ของอาร์เรย์ A[n]

มี i–1 ช่อง คิดเป็นขนาดพื้นที่ = C(i-1) Byte

ซึ่งสามารถเขียนสูตรการคำนวณได้ดังนี้

$$Loc(A[i]) = B + C(i - L)$$

โดยที่ Loc(A[i]) คือ ตำแหน่งแอดเดรส ที่เก็บ A[i] ในหน่วยความจำ

B คือ แอดเดรสเริ่มต้นของอาร์เรย์ A

C คือ ขนาดความกว้างของแต่ละช่องของอาร์เรย์

i คือ ตำแหน่งที่ต้องการหาแอดเดรส

L คือ ขอบเขตล่างสุด

ตัวอย่าง กำหนดอาร์เรย์ A[1:8] สมาชิกแต่ละตัวใช้พื้นที่ในหน่วยความจำ 4 Byte และ A[1] ถูกบันทึกในตำแหน่งแอดเดรสที่ 400 จงคำนวณหาตำแหน่งแอดเดรสของ A[6]

จากสูตร $Loc(A[i]) = B + C(i - L)$

$Loc(A[6] = 400 + 4(6 - 1)$

$= 400 + 4(5)$

$= 420$

นั่นคือ A[6] ถูกบันทึกในตำแหน่งแอดเดรสที่ 420 ในหน่วยความจำ

แอดเดรส	400	404	408	412	416	420	424	428
	1	2	3	4	5	6	7	8
	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]
	4 Byte	4 Byte	4 Byte	4 Byte	4 Byte	4 Byte	4 Byte	4 Byte

แสดงอาร์เรย์ A ที่จัดเก็บในหน่วยความจำคอมพิวเตอร์

ตัวอย่าง กำหนดอาร์เรย์ M[-2:8] สมาชิกแต่ละตัวใช้พื้นที่ในหน่วยความจำ 2 Byte และ M[-2] ถูกบันทึกในตำแหน่งแอดเดรสที่ 250 จงคำนวณหาตำแหน่งแอดเดรสของ M[5]

จากสูตร $Loc(A[i]) = B + C(i - L)$

$Loc(M[5] = 250 + 2(5 - (-2))$

$= 250 + 2(7)$

$= 264$

นั่นคือ M[5] ถูกบันทึกในตำแหน่งแอดเดรสที่ 264 ในหน่วยความจำ