Introduction

02204231 Data Structures and Algorithms I

Computer Engineering, Kasetsart University Kamphaeng Sean Campus

Outline

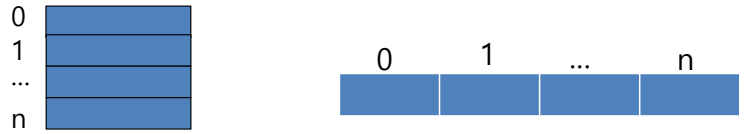
1. Course Overview
   แนะนำรายวิชา
2. Data Structures
   โครงสร้างข้อมูลคืออะไร
3. Algorithm
   ขั้นตอนวิธี
4. Revision of Basic Data Types
   ทบทวนชนิดข้อมูลพื้นฐาน

Revision of Basic Data Types

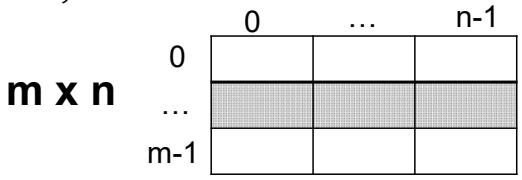
- **อาเรย์ (Arrays)** ตัวแปรชุด ที่เก็บ**ข้อมูลประเภทเดียวกัน**

  อาเรย์ช่วยอำนวยความสะดวกในการประกาศและใช้งานตัวแปรจำนวนมาก

  - Array 1 มิติ เปรียบได้กับ vector คือ มีเพียงแถวหรือคอลัมน์



  - Array 2 มิติ เปรียบได้กับ matrix ที่มีทั้งแถวและคอลัมน์



Revision of Basic Data Types

(Lack of) Aggregate Array Operations

```
int x[10];
int a;

x=5;
a=5;
x[2]=5;
```

Assignment

```
int x[5]={1,2,
3,4,5};
int y[5]={0};

x=x*2;
x=x+y;
```

Arithmetic

```
int x[5];
int y[5];

if(x<y)
printf("Hi");
```

Comparison

```
int x[5]={0};
char z[5]="xx";

printf("%d",x);
printf("%s",z);
```

Basic I/O

Revision of Basic Data Types

(Lack of) Aggregate Array Operations

- เมื่อดำเนินการต่อไปนี้กับอาเรย์จะต้องมีการระบุ index เสมอ
  - การกำหนดค่า (Assignment) ให้กับสมาชิกในอาเรย์
  - การคำนวณทางคณิตศาสตร์ (Arithmetic)
  - การเปรียบเทียบ (Comparison)
  - การรับข้อมูลเข้าและแสดงผล (Basic I/O) –ยกเว้น C string

การดำเนินการเหล่านี้ ไม่สามารถทำกับทั้งชุดของอาเรย์ได้ ต้องดำเนินการกับสมาชิกแต่ละตัวในอาเรย์ (→ วนลูป)

Revision of Basic Data Types

(Lack of) Aggregate Array Operations

Exercise3

เขียนโปรแกรมในภาษา C หรือ Java ตามรายละเอียดต่อไปนี้

1. สร้าง vector X ให้มีค่าเริ่มต้นเท่ากับ {1,2,3,4,5}
2. สร้าง vector Y ให้มีค่าจากการคำนวณ Y= mX+c (m มีค่า 5, c มีค่า 3)
3. แสดงค่า vector Y ทางจอภาพ
4. สร้าง vector Z ให้มีค่าเริ่มต้นเป็น {8,13,18,23,28} แทนผลเฉลยสมการ
5. ตรวจสอบว่า vector Y เท่ากันกับ vector Z หรือไม่

Revision of Basic Data Types

Basic Operations of Array

การดำเนินการ/การทำงานโดยใช้อาเรย์ ได้แก่

- การเข้าถึงสมาชิกทุกตัวในอาเรย์ (Traverse)
  - เพื่อคำนวณ เปรียบเทียบ รับค่าจากคีย์บอรด์ แสดงค่า
- การแก้ไขค่าสมาชิกในอาเรย์ โดยระบุ index (Update)
- การค้นหาค่าในอาเรย์ (Search)
- การเพิ่ม/ลบ สมาชิกในอาเรย์ (Add/Remove)
  - ส่วนหัว/ส่วนท้าย/ระบุ index

Revision of Basic Data Types

Basic Operations of Array

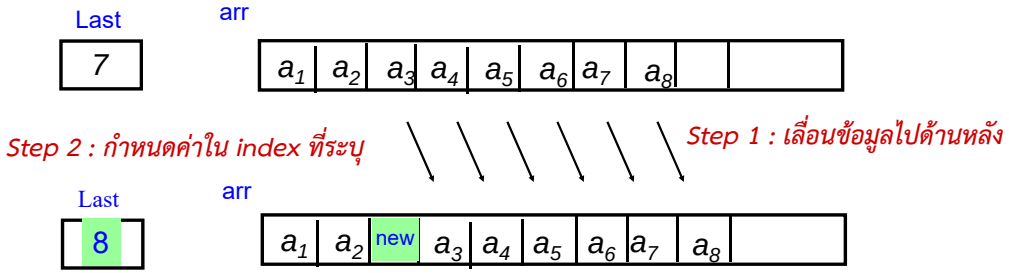
Exercise4

- การเพิ่ม/ลบ สมาชิกในอาเรย์ (Add/Remove) ในตำแหน่งใด ที่ใช้เวลาในการดำเนินการเป็นค่าคงที่ (ไม่ขึ้นกับขนาดของอาเรย์)
  - ส่วนหัว
  - ส่วนท้าย
  - ระบุ index

Revision of Basic Data Types

Basic Operations of Array

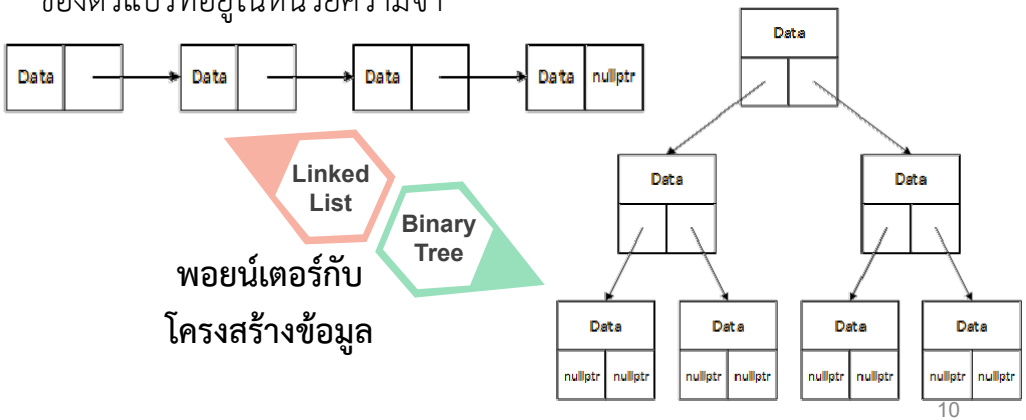
Example : เพิ่มค่าอาเรย์ arr ที่ index 2



Step 3 : ปรับค่า index สุดท้ายของอาเรย์

Revision of Basic Data Types

- **พอยน์เตอร์ (Pointer)** ชนิดตัวแปรที่สำหรับเก็บตำแหน่งที่อยู่ (Address) ของตัวแปรที่อยู่ในหน่วยความจำ



Revision of Basic Data Types

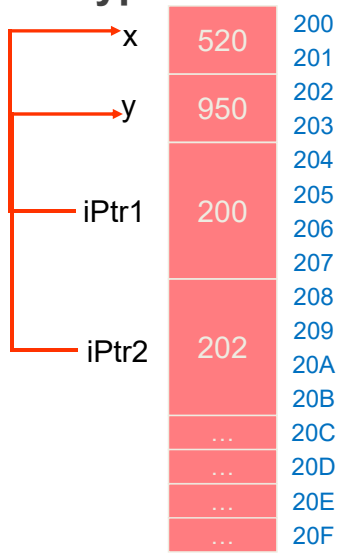
Pointer

```
int x = 520,y = 950;
int *iPtr1,int *iPtr2;

iPtr1 = &x; //iPtr1 = 200
iPtr2 = &y; //iPtr2 = 202
printf("%d\n", *iPtr1);
printf("%d", *iPtr2);

iPtr1 = x; ?
x = iPtr1; ?
y = *iPtr1; ?

printf("%d", y);
```



Revision of Basic Data Types

Pointer

Exercise5 จงแสดงผลลัพธ์ทางจอภาพของโปรแกรมต่อไปนี้

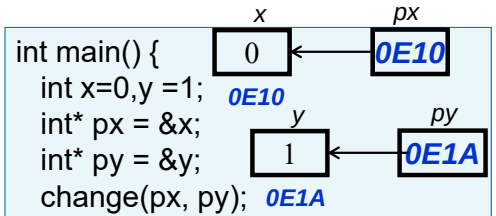
```
void change(int* px, int* py) {
   *px = 5;
    px = py;

    printf("(x, y) = (%d, %d)\n", *px, *py);
}
```

```
int main() {
   int x=0,y =1;
   int* px = &x;
   int* py = &y;
   change(px, py);

  printf("(x, y) = (%d, %d)\n", x, y);

  *px = 2;
   printf("(x, y) = (%d, %d)\n", x, y);
   return 0;
}
```



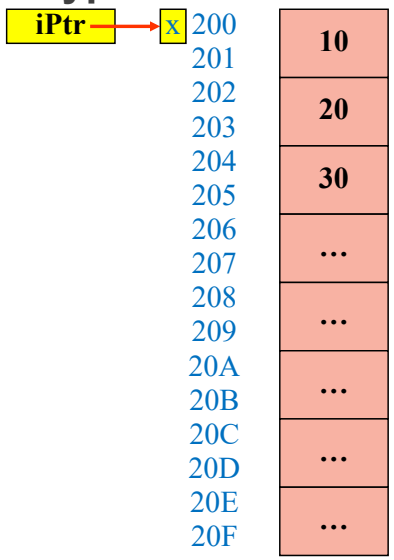
Revision of Basic Data Types

Pointer

Exercise6

```
int x[10]={10,20,30};
int *iPtr = x;
```

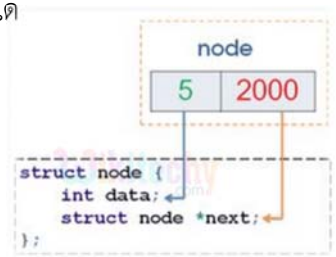
1. iPtr
2. *iPtr
3. iPtr++
4. *(++iPtr)
5. ++(*iPtr)
6. x
7. x[0]
8. &x[0]



Revision of Structure

- สตรัคเจอร์ (Structure) หรือกลุ่มข้อมูลชนิดโครงสร้าง เป็นการกำหนดชนิดของข้อมูล (Data type) ขึ้นมาใหม่
  - การนำชนิดข้อมูลพื้นฐานในภาษาซี เช่น int, char, float มาประกอบกันเป็นโครงสร้างของข้อมูลชนิดใหม่ โดยข้อมูลซึ่งเป็นสมาชิกของโครงสร้างใหม่ อาจมีหลายตัว และเป็นชนิดเดียวกันหรือต่างชนิดกันก็ได้

ตัวอย่าง: การสร้าง Linked List
โครงสร้างข้อมูลแบบลิงค์ลิสต์ ที่มีลักษณะเป็นก้อนข้อมูลชนิดสตรัคเจอร์ และเชื่อมกับก้อนข้อมูลชนิดเดียวกันด้วยพอยน์เตอร์



Revision of Structure

การนิยามและการประกาศสตรัคเจอร์ (Structure definition & declaration)

การนิยามกลุ่มข้อมูลที่สร้างใหม่ว่ามีสมาชิกอะไรบ้าง เป็นชนิดใดเสมือนเป็นการสร้างแบบ (template) ที่จะสามารถนำไปใช้ในการสร้างตัวแปรต่อไป

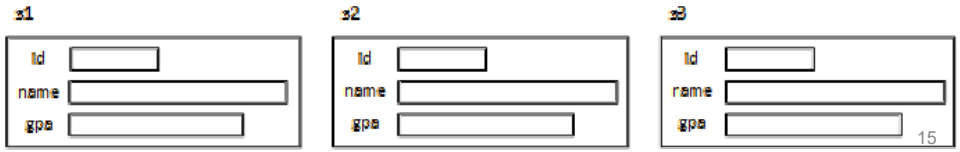
การนิยามสตรัคเจอร์

```
struct student
{
    int     id;
    char    name[20];
    double  gpa;
};
```

```
struct student s1;
struct student s2;
struct student s3;
```

การประกาศตัวแปรให้มีโครงสร้างตามที่กำหนดไว้แล้วโดยใช้รูปแบบต่อไปนี้

การประกาศสตรัคเจอร์



Revision of Structure

การนิยามและการประกาศสตรัคเจอร์ (Structure definition & declaration) ด้วย typedef

การนิยามสตรัคเจอร์ด้วยคีย์เวิร์ด typedef ทำให้การประกาศตัวแปรชนิดสตรัคสั้นลง

```
struct student
{
    int     id;
    char    name[20];
    double  gpa;
};
```

```
struct student s1;
struct student s2;
struct student s3;
```

Code without typedef

```
typedef struct {
    int     id;
    char    name[20];
    double  gpa;
} student;
```

```
student s1;
student s2;
student s3;
```

```
typedef struct student
{
    int     id;
    char    name[20];
    double  gpa;
} std;
```

```
std s1;
struct student s2;
struct student s3;
```

Code using typedef

Revision of Structure

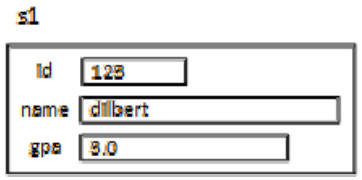
การเข้าถึงสมาชิกในตัวแปรชนิดสตรัคเจอร์

การเข้าถึงสมาชิกในตัวแปรชนิดสตรัคเจอร์ ทำได้โดยบอกชื่อตัวแปรสตรัคเจอร์ ตามด้วยจุด “.” และต่อด้วยชื่อสมาชิกของสตรัคเจอร์นั้นๆ

ตัวแปรสตรัคเจอร์.ชื่อสมาชิกของสตรัคเจอร์;

Assignment

```
s1.id=123;
strcpy(s1.name,”dilbert”);
s1.gpa=3.0;
```




```
printf("%s %f",s1.name,s1.gpa);
```

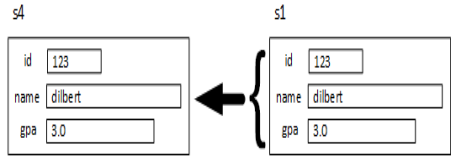
Basic I/O

Revision of Structure

Aggregate Structure Operations

- เมื่อดำเนินการต่อไปนี้กับตัวแปรชนิดสตรัคเจอร์ จะต้องมีการระบุชื่อตัวแปรและชื่อสมาชิกเสมอ เสมอ
  - การกำหนดค่า (Assignment)
    ยกเว้นการกำหนดค่าจากตัวแปรชนิดสตัครเจอร์ (Copy)
  - การคำนวณทางคณิตศาสตร์ (Arithmetic)
  - การเปรียบเทียบ (Comparison)
  - การรับข้อมูลเข้าและแสดงผล (Basic I/O)

```
struct student s4;
s4 = s1;
```



Revision of Structure

พอยน์เตอร์ชนิดสตรัคเจอร์ (Pointers to Structures)

เช่นเดียวกับการใช้พอยน์เตอร์กับตัวแปรชนิดอื่นๆ เราสามารถใช้พอยน์เตอร์ชี้ไปยังสตรัคเจอร์ได้ ซึ่งการใช้งานส่วนใหญ่จะเป็นการผ่านค่าตัวแปรเข้าไปในฟังก์ชัน

Code without typedef

```
struct student *sp;
```

Code using typedef

```
student *sp;
```

การเข้าถึงสมาชิกในตัวแปรชนิดสตรัคเจอร์ ผ่านพอยน์เตอร์ทำได้โดยบอกชื่อตัวแปรพอยน์เตอร์สตรัคเจอร์ ตามด้วย “->” และต่อด้วยชื่อสมาชิกของสตรัคเจอร์นั้นๆ

```
sp = &s4;

printf("%s\n", sp->name);
printf("%s", s4.name);
```

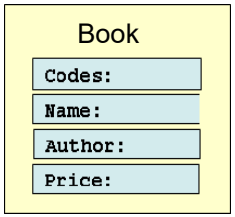

Revision of Basic Data Types

Structure

Exercise7 กำหนดโครงสร้างสำหรับเก็บข้อมูลหนังสือดังรูป จงสร้างชนิดข้อมูลชื่อ Book และดำเนินการกับตัวแปรชนิด Book ที่สร้างขึ้น ตามคำสั่งต่อไปนี้

- เขียนคำสั่งนิยามสตรัคเจอร์ชื่อ Book โดยให้กำหนดชนิดข้อมูลให้เหมาะสม
- เขียนคำสั่งประกาศตัวแปรชื่อ B1 ชนิด Book พร้อมกำหนดค่าเริ่มต้นดังนี้

  Codes: 12345
  Name: The suicide shop
  Author: Jean Teule
  Price: 200



- เขียนคำสั่งประกาศตัวแปรชื่อ BookList[5] ชนิด Book
- Copy ค่า ของตัวแปร B1 ให้กับตัวแปร BookList ตัวสุดท้าย