แผนการสอนประจำสัปดาห์ที่ 9

หัวข้อเนื้อหา โครงสร้างควบคุม (Control Structure)

รายละเอียด

ศึกษาเกี่ยวกับการใช้คำสั่งกระโดดมาสร้างเป็นโครงสร้างควบคุมในรูปแบบต่าง ๆ การใช้คำสั่งกระโดดในลักษณะต่างๆ จะทำให้ชุดคำสั่งมีความเป็นโครงสร้างมากขึ้น

จำนวนชั่วโมงที่สอน 3 ชั่วโมง/สัปดาห์

กิจกรรมการเรียนการสอน

1. บรรยาย
2. สืบเสาะหาความรู้
3. ค้นคว้าเพิ่มเติม
4. ตอบคำถาม

สื่อการสอน

1. สื่ออิเล็กทรอนิกส์
2. เพาเวอร์พอยต์ พรีเซนเตชัน
3. บทเรียนออนไลน์
4. เอกสารอ้างอิงประกอบการค้นคว้า

แผนการประเมินผลการเรียนรู้

1. ผลการเรียนรู้

1.1 สังเกตจากงานที่กำหนดให้ไปทำมาส่ง

1.2 สังเกตจากการตอบคำถาม

1.3 สังเกตจากการนำความรู้ไปใช้

2. วิธีการประเมินผลการเรียนรู้

2.1 ตรวจผลงานภาคปฏิบัติ

2.2 ตรวจรายงาน

2.3 ตรวจแบบฝึกหัด

3. สัดส่วนของการประเมิน

3.1 ใบงานที่นักศึกษาทำมาส่ง

3.2 คะแนนเก็บในชั้นเรียน

3.3 การเข้าชั้นเรียน

เนื้อหาที่สอน

ในสัปดาห์ที่ 9 การจัดการเรียนการสอน จะเกี่ยวข้องกับโครงสร้างการตัดสินใจแบบ if-then-else การสร้าง repeat until loop การสร้าง while loop และการสร้าง for loop ซึ่งคำสั่งกระโดด ทำให้สามารถเขียนโปรแกรมให้มีการทำงานที่ซับซ้อนขึ้นได้ แต่การใช้กลุ่มคำสั่งต่างๆ อย่างไม่เป็นระบบทำให้ชุดคำสั่งที่เขียนขึ้นมานั้นทำความเข้าใจได้ยากและมีลักษณะไม่เป็นระบบที่สมบูรณ์ ดังนั้นการใช้คำสั่งกระโดดมาสร้างเป็นโครงสร้างควบคุมในรูปแบบต่าง ๆ จะทำให้ชุดคำสั่งมีความเป็นโครงสร้างมากขึ้น (ศัพท์บัญญัติ ราชบัณฑิตยสถาน, 2544)

9.1 การสร้างโครงสร้างการตัดสินใจแบบ if-then-else

รูปแบบของโครงสร้างที่ง่ายที่สุดคือโครงสร้างแบบ if-then-else รูปแบบของโปรแกรมภาษาแอสแซมบลีให้มีโครงสร้างแบบ if-then-else มีลักษณะดังนี้

```
if condition is false  then jump to elselabel
        then_actions
        jump to endif_label
elselabel:
        else_actions
endif_label:
```

ตัวอย่าง โปรแกรมที่ 9.1 เปรียบเทียบการเขียนโปรแกรมด้วยโครงสร้างของภาษา pascal กับโครงสร้างภาษา Assembly

(a)

```
if AL<10 the

DL:=AL+'0'

        else
  DL:=AL+'A'-10;
```

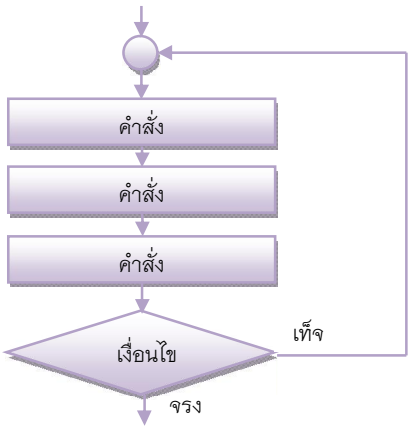
(b)

```
        cmp   al,10
        jae   abovenine
        mov   dl,al
        add   dl,'0'
        jmp   endif
abovenine:
        mov   dl,al
        add   al,'A'-10
endif:
```

โปรแกรมที่ 9.1 โปรแกรม (a) แสดงตัวอย่างของโปรแกรมที่เขียนด้วยโครงสร้างของภาษา pascal โปรแกรม (b) แสดงโปรแกรมภาษาแอสแซมบลี (Assembly) ที่เทียบเท่ากัน

9.2 การสร้าง repeat until loop

โครงสร้างของ repeat until loop ในภาษาระดับสูงทั่วไปมีลักษณะ แสดงได้ดังภาพที่ 9.1



ภาพที่ 9.1 แสดงโครงสร้างควบคุมแบบ repeat until

ตัวอย่าง โปรแกรมที่ 9.2 การเขียนโปรแกรมโดยใช้ภาษาแอสแซมบลีโดยมีโครงสร้างแบบ repeat until

(a)

```
BL:=1;
CX:=0;
DX:=0;
Repeat
DX:=DX+BL*BL;

BL:=BL+1;
CX:=CX+1;
until (DX>100);
```

(b)

```
        mov   bl,1
        mov   cx,0
        mov   dx,0
startloop:
        mov   al,bl
        mul   bl    ;result in ax
        add   dx,ax
        inc   bx
        inc   cx
        cmp   dx,100
        jbe   startloop
```

โปรแกรมที่ 9.2 โปรแกรม (a) แสดงตัวอย่างของโปรแกรมภาษา Pascal ที่ใช้โครงสร้างแบบ repeat until
โปรแกรม (b) แสดงโปรแกรมภาษาแอสแซมบลี (Assembly) ที่เทียบเท่ากัน

รูปแบบของโปรแกรมภาษาแอสแซมบลีที่เทียบเท่ากับ repeat until loop มีลักษณะเป็นดังนี้

startlabel:

action;

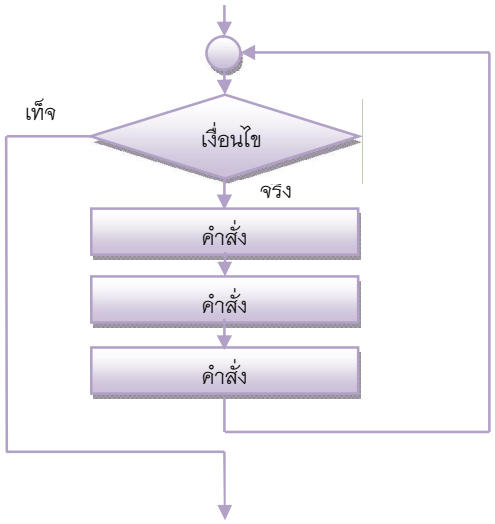
...

action;

if *condition is false* then jump to startlabel

9.3 การสร้าง while loop

โครงสร้างของ while loop ในภาษาระดับสูงทั่วไป แสดงได้ดังภาพที่ 9.2



ภาพที่ 9.2 โครงสร้างควบคุมแบบ while loop

ตัวอย่าง โปรแกรมที่ 9.3 การเขียนโปรแกรมภาษาแอสแซมบลีโดยมีโครงสร้างแบบ while loop

(a)

```
while(DL<>13) and
      (CX<20)  do

begin
   AX:=AX+DL;

   BX:=BX+1;
   DL:=DATA[BX];
   CX:=CX+1;

end;
```

(b)

```
startloop:
cmp   dl,13
jz    endloop
cmp   cx,20
jae   endloop
add   al,dl
adc   ah,0
inc   bx
mov       dl,data[bx]
inc   cx
jmp   startloop
endloop:
```

โปรแกรม (a) แสดงตัวอย่างของโปรแกรมภาษา pascal ที่ใช้โครงสร้างแบบ while loop

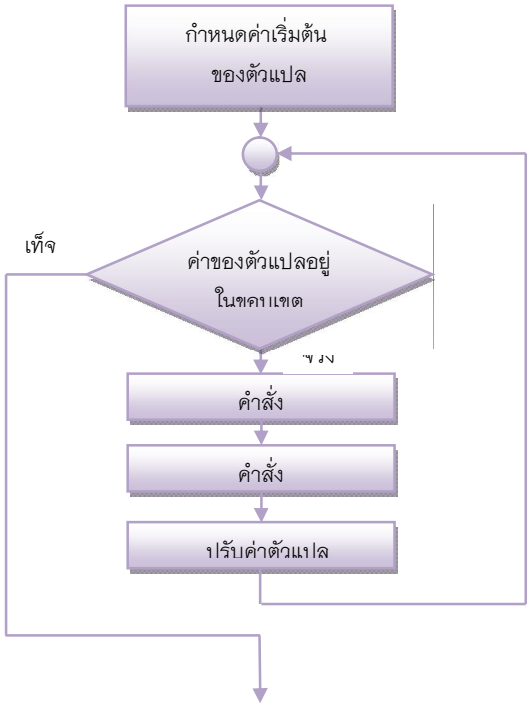
โปรแกรม (b) แสดงโปรแกรมภาษาแอสแซมบลีที่เทียบเท่ากัน

รูปแบบของโปรแกรมภาษาแอสแซมบลีที่เทียบเท่ากับ while loop มีลักษณะเป็นดังนี้

startlabel:
if *condition is false* then
    jump to endlabel
        action
        ...
        action
    jump to startlabel
endlabel:

9.4 การสร้าง for loop

เราสามารถใช้คำสั่งกระโดดในการสร้างโครงสร้างแบบ for loop ได้ นอกจากนั้นเรายังสามารถใช้คำสั่งกลุ่ม LOOP ในการสร้างโครงสร้างแบบ for loop ได้เช่นเดียวกัน โครงสร้างของ for loop มีลักษณะ แสดงได้ดังภาพที่ 9.3



ภาพที่ 9.3 โครงสร้างของ for loop

รูปแบบของโปรแกรมภาษาแอสแซมบลีที่มีโครงสร้างเป็นแบบ for loop มีลักษณะดังนี้

initialize index variables

startloop:

if *index value is not in the range* then jump to endloop

action

. . .

action

update index variable

jump to startloop

endloop:

ตัวอย่าง โปรแกรมที่เขียนโดยใช้รูปแบบลักษณะนี้เป็นดังโปรแกรมที่ 9.4

(a)

```
CX:=0;

for DL:=1 to 100 do

  begin
 if DL mod 7=0 then

     CX:=CX+1;

 end;
```

(b)

```
        mov   cx,0

        mov   dl,1
startloop:
        cmp   dl,100
        ja    endloop

        mov   al,dl
        mov   ah,0
        mov   bl,7
        div   bl

        cmp   ah,0
        jne   endif

        inc   cx
endif:

        inc   dl
        jmp   startloop
endloop:
```

โปรแกรมที่ 9.4 โปรแกรม (a) แสดงโปรแกรมที่เขียนด้วยภาษา pascal ที่ใช้โครงสร้างแบบ for loop
โปรแกรม (b) แสดงโปรแกรมที่เขียนด้วยภาษาแอสแซมบลี

นอกจากนี้ยังสามารถใช้คำสั่ง LOOP ในการสร้างโครงสร้างแบบ for loop ได้เช่นเดียวกัน ดังต่อไปนี้

```
        set the value of CX
startloop:
        actions
        LOOP startloop
```

แต่การใช้คำสั่ง LOOP ในการสร้างโครงสร้างแบบ for loop ไม่สามารถสร้างโครงสร้างการกระทำซ้ำที่มีความซับซ้อนมาก ๆ ได้ เช่นการกระทำซ้ำที่มีวงรอบของการกระทำซ้ำซ้อนกันหลาย ๆ วง

สำหรับคำสั่ง LOOPZ และ LOOPNZ นั้น เราสามารถนำมาใช้ในการสร้างโครงสร้างควบคุมที่มีความซับซ้อนขึ้นได้ โดยโครงสร้างดังกล่าวจะมีลักษณะปนกันระหว่าง for loop และ while loop หรือ repeat until นั่นคือเงื่อนไขควบคุมการกระทำซ้ำจะขึ้นกับทั้งค่าของรีจิสเตอร์ (มีลักษณะคล้ายโครงสร้างแบบ for loop) และเป็นเงื่อนไขจริง ๆ (คล้าย repeat until loop และ while loop) ดังในตัวอย่างโปรแกรมต่อไปนี้

(a)

```
AX:=0;
CX:=100;

repeat
   AX:=AX+data[BX];
   BX:=BX+2;
   CX:=CX-1;
until (data[BX]=0) or
      (CX=0);
```

(b)

```
      mov   ax,0
      mov   cx,100

startloop:
      add   ax,data[BX]
      add   bx,2
      cmp   data[BX],0
      loopnz startloop
```

โปรแกรมที่ 9.5 โปรแกรม (a) แสดงโปรแกรมที่เขียนด้วยภาษา pascal
โปรแกรม (b) แสดงโปรแกรมที่เขียนด้วยภาษาแอสแซมบลี

สรุป

การใช้คำสั่งกระโดดมาสร้างเป็นโครงสร้างควบคุมรูปแบบต่าง ๆ การใช้คำสั่งกระโดดในลักษณะนี้จะทำให้โปรแกรมของเรามีความเป็นโครงสร้างมากขึ้น เช่น การสร้างโครงสร้างการตัดสินใจแบบ if-then-else, repeat until loop, while loop และ for loop เป็นต้น ดังนั้นผู้เขียนโปรแกรมภาษาแอสแซมบลีจำเป็นจะต้องทำความเข้าวิธีการและรูปแบบการเลือกการสร้างโครงสร้างควบคุมใช้งานให้เหมาะสมมากที่สุด

คำถามทบทวน

1. อธิบายพร้อมยกตัวอย่างการทำงานชองโครงสร้างควบคุมรูปแบบ if-then-else
2. อธิบายพร้อมยกตัวอย่างการทำงานชองโครงสร้างควบคุมรูปแบบ repeat until loop
3. อธิบายพร้อมยกตัวอย่างการทำงานชองโครงสร้างควบคุมรูปแบบ while loop
4. อธิบายพร้อมยกตัวอย่างการทำงานชองโครงสร้างควบคุมรูปแบบ for loop
5. จงเปรียบเทียบข้อดีและข้อเสียของรูปแบบโครงสร้างควบคุมแบบ repeat until และ while loop
6. จงเปลี่ยนโปรแกรมที่เขียนด้วยภาษาโปรแกรม pascal ที่ให้มาให้อยู่ในรูปของโปรแกรมภาษาแอสแซมบลีที่เทียบเท่ากัน

```
AX:=20;
CX:=100;
repeat
   AX:=AX+data[BX];
   BX:=BX+8;
   CX:=CX-2;
until (data[BX]=0) or
   (CX=0);
```

เอกสารอ้างอิง

ราชบัณฑิตยสถาน. (2544). *ศัพท์บัญญัติ ราชบัณฑิตยสถาน.* ค้นเมื่อ 9 มีนาคม 2557,จาก :http://rirs3.royin.go.th/coinages/

โครงสร้างควบคุม. (2557). *วิกิพีเดีย สารานุกรมเสรี.* ค้นเมื่อ 9 มีนาคม 2557, จาก: http://th.wikipedia.org/wiki/

ชูชัย ธนสารตั้งเจริญ, กำธร พานิชปฐมพงษ์. *ภาษาแอสแซมบลี 80286/80386(PC).* กรุงเทพฯ :สำนักพิมพ์ซีเอ็ดยูเคชั่น บมจ, 2536.

ธีรวัฒน์ ประกอบผล. *ระบบคอมพิวเตอร์และภาษาแอสแซมบลี.* กรุงเทพฯ :สำนักพิมพ์ส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2537.