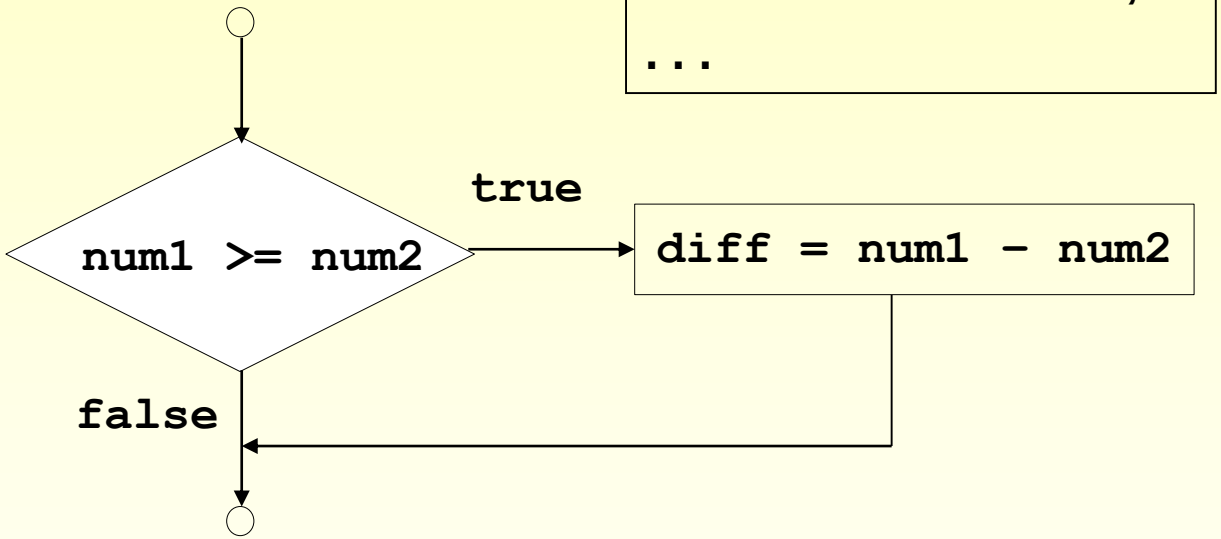
C Programming
Lecture 7 : Control Structures

Control Structures

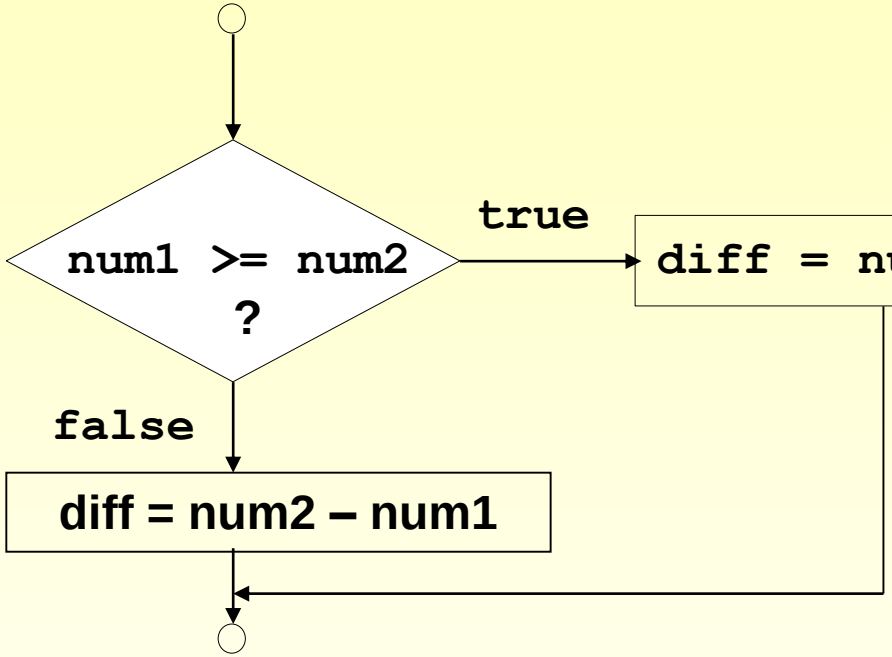
- **Conditional statement : `if, switch`**
  - Determine a block of statements to execute depending on whether the condition is true or false
- **Repetition statement : `for, while, do-while`**
  - Loop : repeat a block of statements a number of times
  - Conditional loop : repeat while the condition is true
- **Other control structures : `goto`, …**

if

```
...
if (num1 >= num2)
   diff = num1 – num2;
...
```



if-else




```
. . .
if (num1 >= num2)
   diff = num1 – num2;
else
   diff = num2 – num1;
. . .
```

if-else

```
if ( grade >= 90 )          // 90 and above
   printf("A");
else if ( grade >= 80 )     // 80-89
   printf("B");
else if ( grade >= 70 )     // 70-79
   printf("C");
else if ( grade >= 60 )     // 60-69
   printf("D");
else                        // less than 60
   printf("F");
```

if example

```
#include <stdio.h>

int main ( )
{
    int num1, num2, num3, min = 0;

    printf (“input three integers : ");
    scanf("%d %d %d", &num1, &num2, &num3);

    if (num1 < num2)
          if (num1 < num3)
                    min = num1;
          else
                    min = num3;
    else
          if (num2 < num3)
                    min = num2;
          else
                    min = num3;

    printf (“min value: %d", min);

    return 0;
}
```

Compound statement

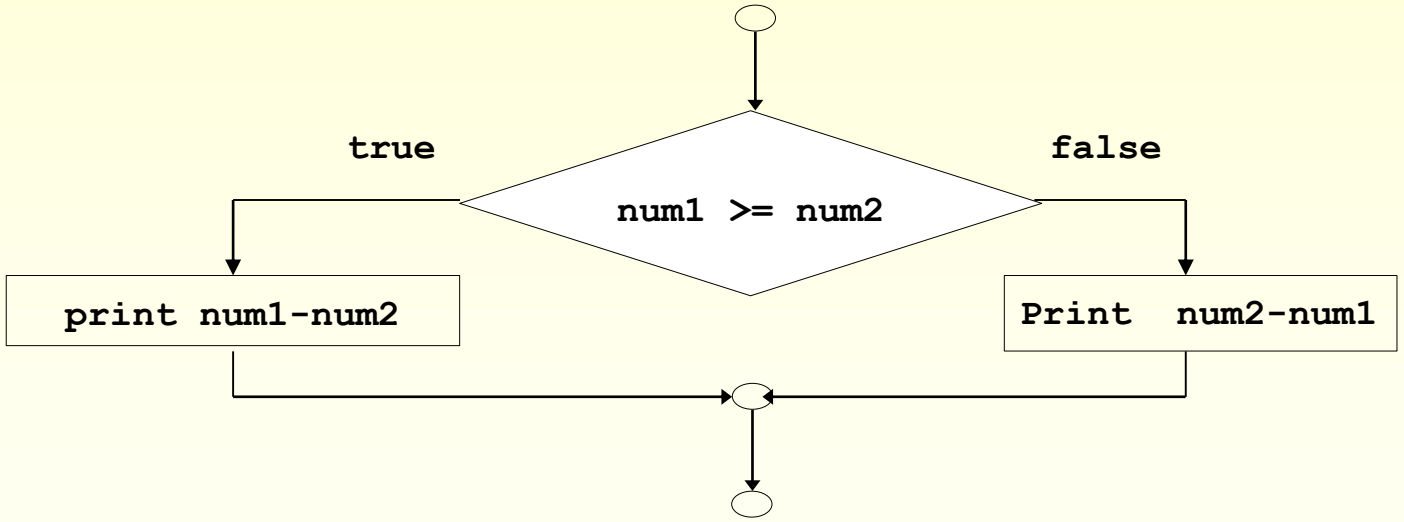
- block : enclosed by { }
- Example

```
if ( num1 >= num2 )
{
     printf("num1 is greater than num2\n");
     printf("The difference is: %d\n", num1- num2);
} else {
     printf("num2 is greater than or equal to num1\n";
     printf("The difference is: %d\n", num2 – num1);
}
```

Ternary conditional operator ?:

- Example

```
printf("Enter two integers :");
scanf("%d %d",&num1, &num2);
printf("%d\n", ((num1 >= num2)? num1-num2: num2-num1));
```



Dangling Else Problem

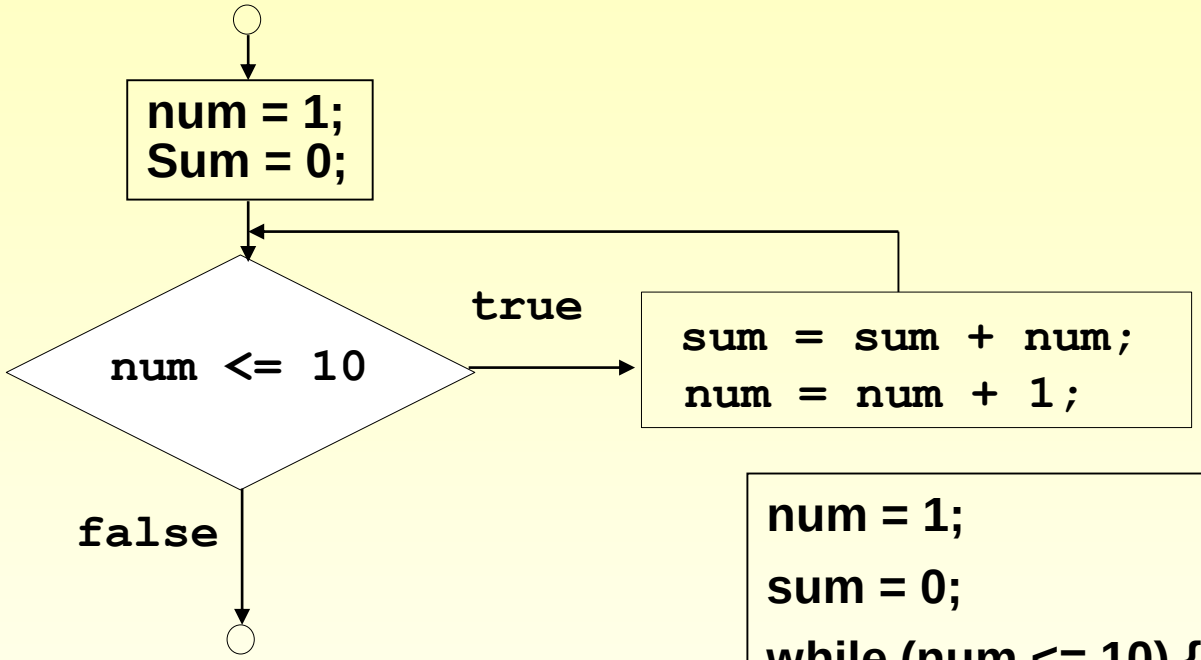
```
if a then if b then s1 else s2
```

switch

- **The `value` in `switch` statement has many cases.**

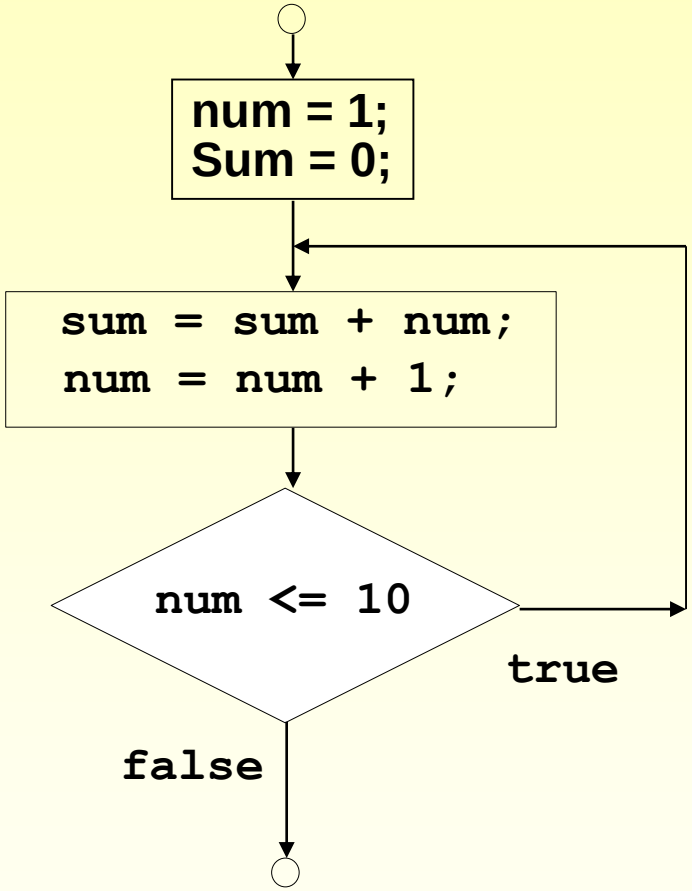
```
int main()
{
   int value;
   scanf("%d",&value);
   switch (value) {
        case 1 :
                printf("1 received\n");
                break;
        case 2 :
                printf("2 received\n");
                break;
        default :
                printf(" values except 1 and 2 were received.\n");
                break;
   }
   return 0;
}
```

while




```
num = 1;
sum = 0;
while (num <= 10) {
    sum = sum + num;
    num = num + 1;
 }
```

do-while



The body (block) of do-while statement is executed at least once.

```
num = 1;
sum = 0;
do {
    sum = sum + num;
    num = num + 1;
 } while (n <= 10)
```

while example

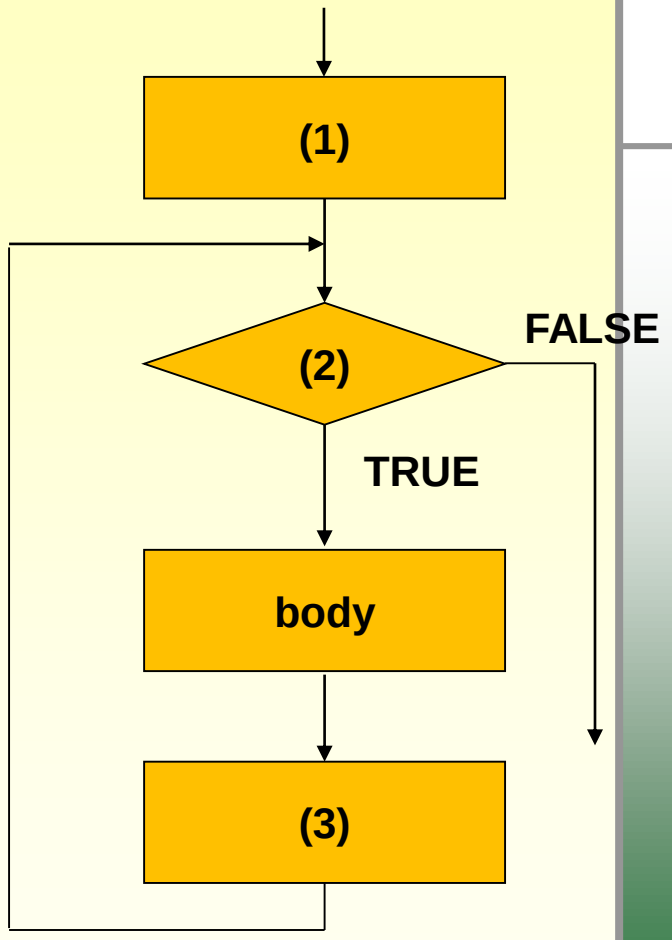
```
#include <stdio.h>

int main ()
{
    int total = 0, score, count = 0;
    float average;

    printf ("score input (quit:0): \n");
    scanf("%d",&score);

    while (score != 0) {
                    total += score;
                    count++;
                    scanf("%d",&score);
    }

    if (count == 0)
          printf ("No input received!");
    else {
                    average = (float) total / count;
                    printf ("total: %d \n", total);
                    printf ("average: %5.2f \n", average);
    }

    return 0;
}
```

for

- Repetition

```
for (   (1);     (2);    (3)    )
{          //   for-repetition body
           ....
           .... // {} is not necessary
              // if there is only one statement in body
}
```

(1) control variable initialization
(2) Test Conditon
(3) Modification of control variable value



order : (1) (2) body (3) (2) body (3) (2) body … body (3) (2)

* Example

```
for(counter = 1; counter <= 10; counter++ )
   printf("%d\n",counter);
```

for example

```
#include <stdio.h>

int main ()
{
    int total = 0, score, count;
    float average;

    printf ("score input (quit:0): ");
    scanf("%d",&score);

    for (count=0; score != 0; count++) {
                    total += score;
                    scanf("%d",&score);
    }

    if (count == 0)
                    printf ("No input received!");
    else {
                    average = (float) total / count;
                    printf ("total: %d \n", total);
                    printf ("avarage: %5.2f \n", average);
    }

    return 0;
}
```

break

- `break` in loop
  - Go out of the loop block and execute next to the loop
  - example

```
while (1) {
     scanf("%d",&j)
    if (j == 0)
         break;
     result = i/j;
}
```

continue

- `continue` in loop
  - Go to condition test of the loop
  - Example

```
for (i = 0, sum = 0; i <= n; i++) {
    if (i % 2 == 0)
        continue;
    sum += i;
}
```

Nested Loop

- loop in a loop

```
int main ()
{
        int i, j;

        for (i=1; i<10; i++) {
                printf ("%d-th iteration  \n", i);
                for (j = 1; j < 10; j++)
                        printf("%d X %d = %d\n", i, j, i*j);
                printf ("\n", i);
        }

        return 0;
}
```

Infinite Loop

- If the condition of the loop is always TRUE, the body of the loop is executed infinitely

- example

```
while(1) {
   i=0;
   i++;
   printf("%d",i);
}
```

```
int count = 1;

while (count != 100)
    count += 2;
```