Problem Solving Through Programming in C

Tutorial Session 3

Prof. Anupam Basu
Dept. of Computer Science & Engg.
IIT Kharagpur

Siddhant Mohapatra
PMRF Scholar
IIT Madras

if-else statement (contd.)

Q. What is the output of the following C code?

```
#include <stdio.h>
int main()
  {
  int x = 1;
  if (x == 0)
    if (x >= 0)
      printf("x=0\n");
    else
      printf("x=1\n");
  return 0;
  }
```

a) x=1
b) x=0
c) Depends on compiler
d) No output

Q. What will be printed when the following C code is executed?

```
#include<stdio.h>
int main()
{
  if('A'<'a')
  printf("NPTEL");
  else
  printf("PROGRAMMING");
  return 0;
}
```

a) NPTEL
b) PROGRAMMING
c) No output
d) Compilation error as 'A' and 'a' are not declared

if-else statement (contd.)

Q. What will be output of the following program?

```
#include<stdio.h>
int main(){
   int a=0,b=10;
   if(a=0){
      printf("true");
   }
   else{
      printf("false");
   }
   return 0;
}
```

a) true
b) false
c) 0
d) 1

Q. What will be the output of the following code

```
int main()
{
 int x, y = 5, z = 5;
 x = y==z;
 printf("%d", x);
 return 0;
}
```

a) 5
b) 1
c) 0
d) None of these

if-else statement (contd.)

Q. What is the output of the following C code?

```
#include <stdio.h>
int main()
{
   int a = 1;
   if (a--)
      printf("True\n");
   if (++a)
      printf("False\n");
  return 0;
}
```

a) True
b) False
c) True
   False
a) Compilation error

postfix operation
↳ return value is the original value

prefix operation
↳ return value is the modified value

assignment
assignment
x = y = z
(R → L) associativity
= operator
y = z
x = y

printf function

What will be the output?

```
#include <stdio.h>
int main()
{
        int x=0;
        x = printf("3");
        printf("%d",x);
        return 0;
}
```

a) 11
b) 33
c) 31
d) 13

What will be the output? (&& is logical AND operation)

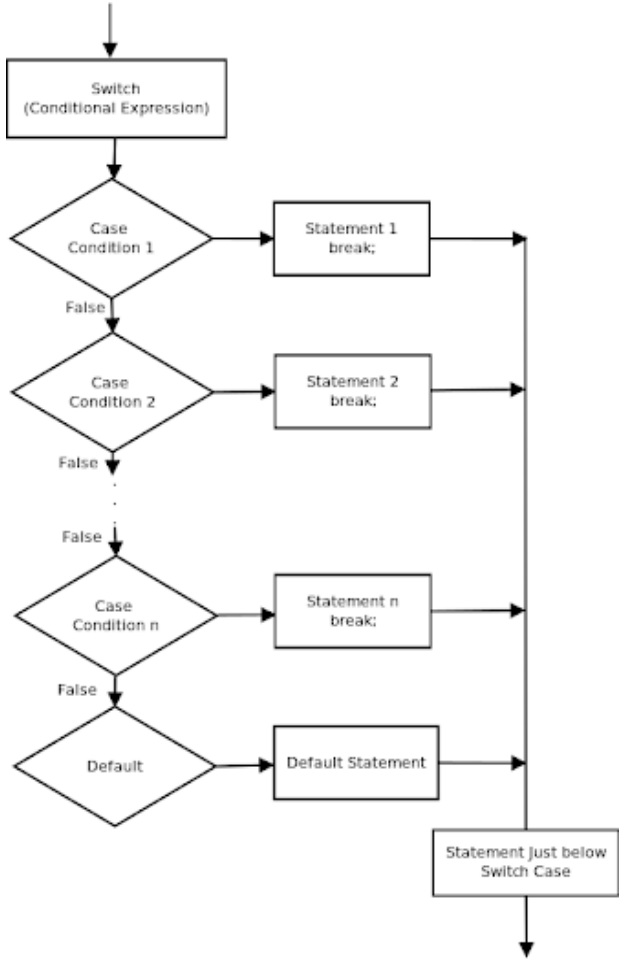
```
#include <stdio.h>
int main()
{
   int i=0, j=1;
   printf("\n %d", i++ && ++j);
   printf("\n %d %d", i,j);
   return 0;
}
```

a) 0
   1 2
b) 1
   1 1
c) 0
   0 0
d) 0
   1 1

switch statement

Syntax:

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```



Q. What will be the output of the following program?

```
#include <stdio.h>
int main()
{
  int x = 1;
  switch (x)
  {
     case 1: printf("Choice is 1 \n");  break;
     default: printf("Choice other than 1 \n");
  }
  return 0;
}
```

a) Choice is 1
b) Choice other than 1
c) Both (a) and (b)
d) Syntax error

Courtesy: w3schools.com; onlinegdb.com

```
y = (b<a) ? ((a<c) ? a-b : b+a) : b-c;
```

```
if (b<a)
{
    if (a<c)
    {
        y = a-b;
    }
    else
        y = b+a;
}
else
    y = b-c;
```

switch statement

Q. Which statement is correct for the following lines?

```
switch(temperatureInCelsius)
    {
      case < 35:
            printf("hot day!");
    default:
            printf("not that hot day!");
    }
```

a) It's perfectly fine
b) It will print both statements
c) It will give a compilation error
d) More information required

Q. What will be the output of the given program?

```
#include <stdio.h>
void main()
{
      float num=5.6;
      switch(num){
            case 5:printf("5");
            case 6:printf("6");
            default : printf("0");
                  break;

      }
      printf("%d", num);
}
```

a) 5 5.600000
b) 6 5.600000
c) 5 6 0 5.600000
d) Compiler error

Loop statements

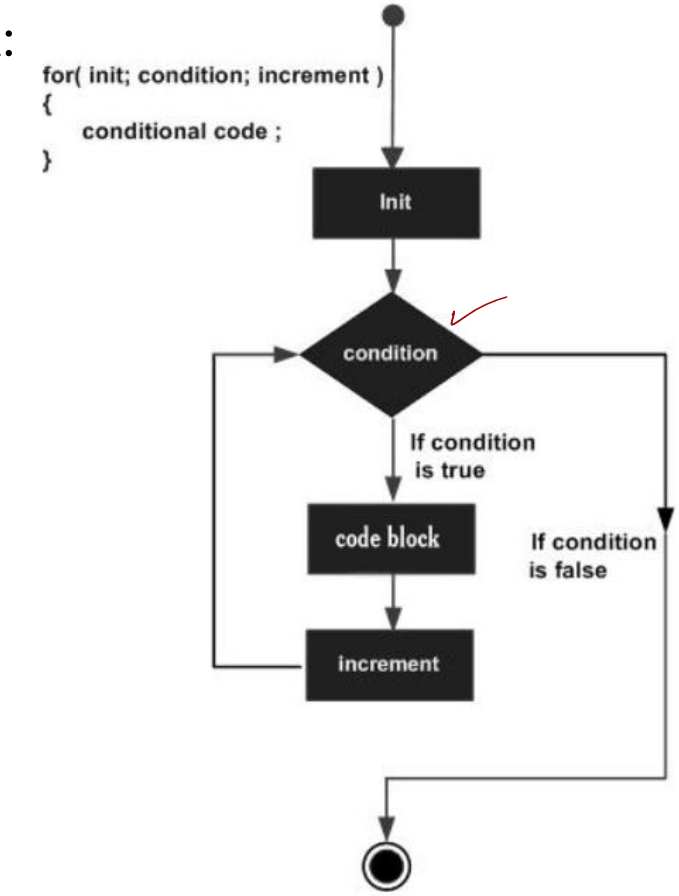
- **for loop**
- while loop
- do-while loop

Example:

```
/*To print all even integers
from 0 to 10 */
#include <stdio.h>

int main() {
    for(int i=0; i<=10; i+=2)
    {
        printf("%d\n", i);
    }
    return 0;
}
```

Syntax:



Courtesy: tutorialspoint.com

Loop statements

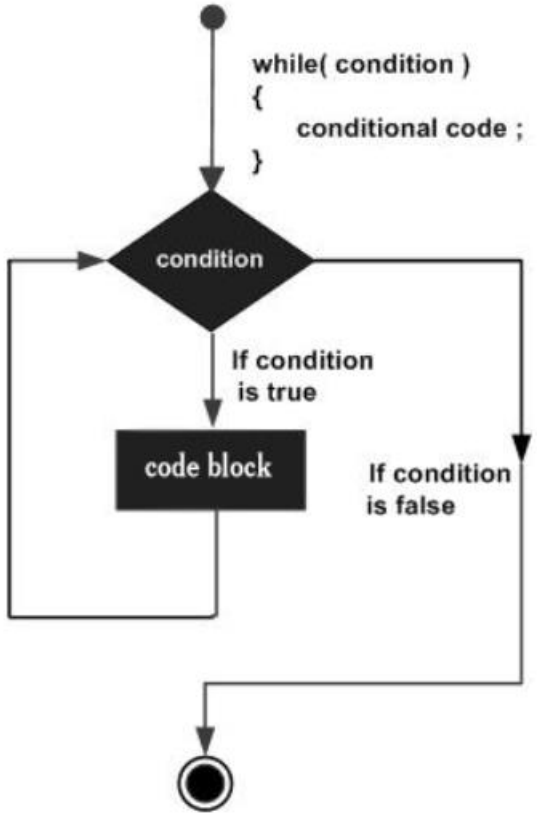
- for loop
- **while loop**
- do-while loop

Example:

```
/*To print all even
integers from 0 to 10 */
#include <stdio.h>

int main() {
    int i=0;
    while(i<=10){
        printf("%d\n",i);
        i+=2;
    }
    return 0;
}
```

Syntax:



Courtesy: tutorialspoint.com

Loop statements

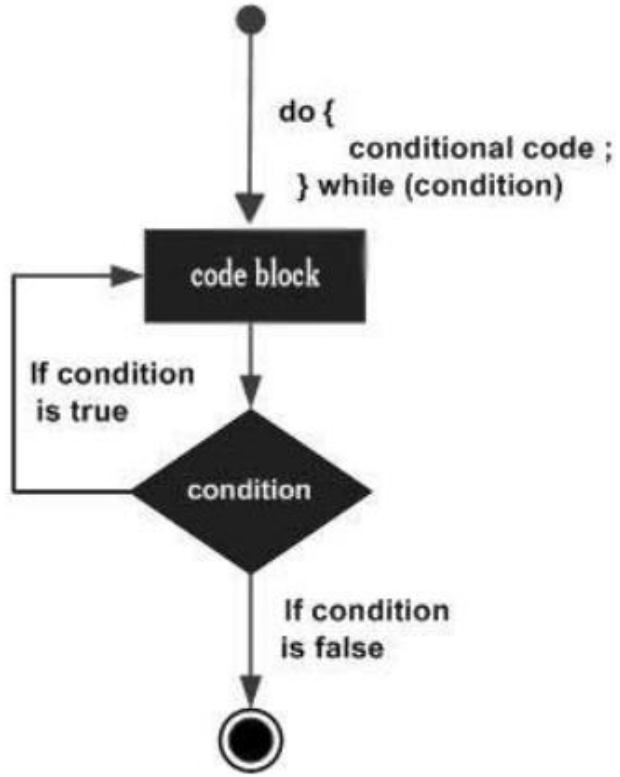
- for loop
- while loop
- **do-while loop**

Example:

```
/*To print all even
integers from 0 to 10 */
#include <stdio.h>

int main() {
    int i=0;
    do {
        printf("%d\n",i);
        i+=2;
        } while(i<=10);
    return 0;
}
```

Syntax:



Courtesy: tutorialspoint.com

Loop statements

Q. What is the purpose of the given program? n is the input number given by the user.

```
#include <stdio.h>
int main()
  {
   int n, x = 0, y;
   printf("Enter an integer: ");
   scanf("%d", &n);
   while (n != 0)
     {
      y = n % 10;
      x = x - y;
      n = n/10;
     }
   printf("Output is = %d", x);
   return 0;
}
```

a) Sum of digits of a number
b) Negative of (a)
c) Reverse of a number
d) Same number is printed

Q. Compute the printed value of i of the C program given below

```
#include<stdio.h>
int main()
{
    int i=2;
    i=i++;
    printf("%d", i);
     return 0;
}
```

a) 2
b) 3
c) 4
d) Compilation error

Loop statements

Q. Write a code to print the Fibonacci sequence up to N terms (0, 1, 1, 2, 3, 5, 8, 13…).

```
#include <stdio.h>
int main()
{
    int x = 0, y = 1, N = 10;
    printf("Fibonacci numbers: %d, %d, ", x, y);
    for(int i = 3; i <= N; ++i)
    {
        y = x+y;
        x = y-x;
        printf("%d, ", y);
    }
    return 0;
}
```

0, 1, 1, 2, 3, 5, ...

1st iteration
y = 0+1 = 1
x = 1-0 = 1

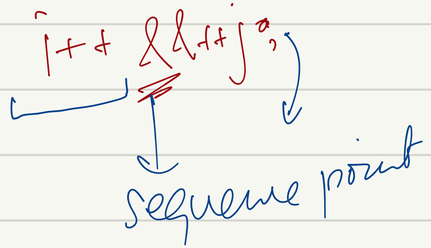
2nd iter
y = 1+1 = 2
x = 2-1 = 1

3rd iter
y = 1+2 = 3
x = 3-1 = 2

4th iter
y = 2+3 = 5
x = 5-2 = 3

tmp = x+y;
printf("%d", tmp);
x = y;
y = tmp;
x, y, tmp

$i = 0, j = 1$



$0 \;\&\&\; = 1\;0$

i++

,

$i = 1$

f(i)

return value = 2

$i = 1$

i++

return value = 1

f(i++)

f(1) $\quad i = 2$

$x = (\underbrace{E}_{exp1}, \underbrace{G}_{exp2}, \underbrace{F}_{exp3})$