BITS Pilani
Pilani Campus

Module 6 – Control Flow: Branching and Looping – Part 1

Dr. Jagat Sesh Challa

Department of Computer Science & Information Systems

Module Overview

- Need for additional programming constructs
- Control Flow: Branching
  - Branching: if statement
  - Branching: if ladder
  - Branching: if-else statement
  - Branching: if-else ladder
  - Branching: nested if-else statement
- Control Flow: Ternary Operator
- Control Flow: Switch Statement

BITS Pilani
Pilani Campus

Need for additional Programming Constructs

Conditionals, Statements and Blocks

Fibonacci Numbers Definition

Consider the following mathematical definition:

$$F_n = \begin{cases} 0 & if\ n = 0 \\ 1 & if\ n = 1 \\ F_{n-1} + F_{n-2} & if\ n > 1 \end{cases}$$

- What additional constructs do we need to define such things in C language?
  - ○ A way to specify the **condition**
    - – Example: $if\ n = 0$
  - ○ A way to selectively choose different blocks of statements depending on the **outcomes of the condition** check
    - – Example: $F_n = 0$ when $n = 0$ and $F_n = 1$ when $n = 1$

Statements and Blocks

Statement:

An expression followed by a semi-colon

Examples: `c = a + b;` `i++;` `Printf(“Hello World”);`

Block:

Set of statements enclosed inside a set of braces { and }

Example:

```
{
        c = a+b;
        c++;
        printf(“c is %d”, c);
}
```

innovate achieve lead

BITS Pilani
Pilani Campus

Control Flow: Branching

Control Flow: Branching

- Different set of instructions gets executed depending on the outcome of a conditional expression

- Writing conditional expression
  - Using relational operators such as ==, >= , <=, != , <, >
  - Using logical operators and : &&, ||, !

- Examples of conditional expressions:

```
(x+y >= 10)
(marks >= 90 && marks <= 100)
(no_of_transaction >= 5 && city == “Metropolitan”)
```

Control Flow: Branching

- Outcome of the condition
  - Non- Zero or ***true***
  - Zero or ***false***

Examples:

```
int x = 5, y = 10;
(x + y <= 20)
int Marks = 95;
(Marks <= 100 && Marks >= 90)
(x & y)
```

Branching: if Statement

Branching: if statement

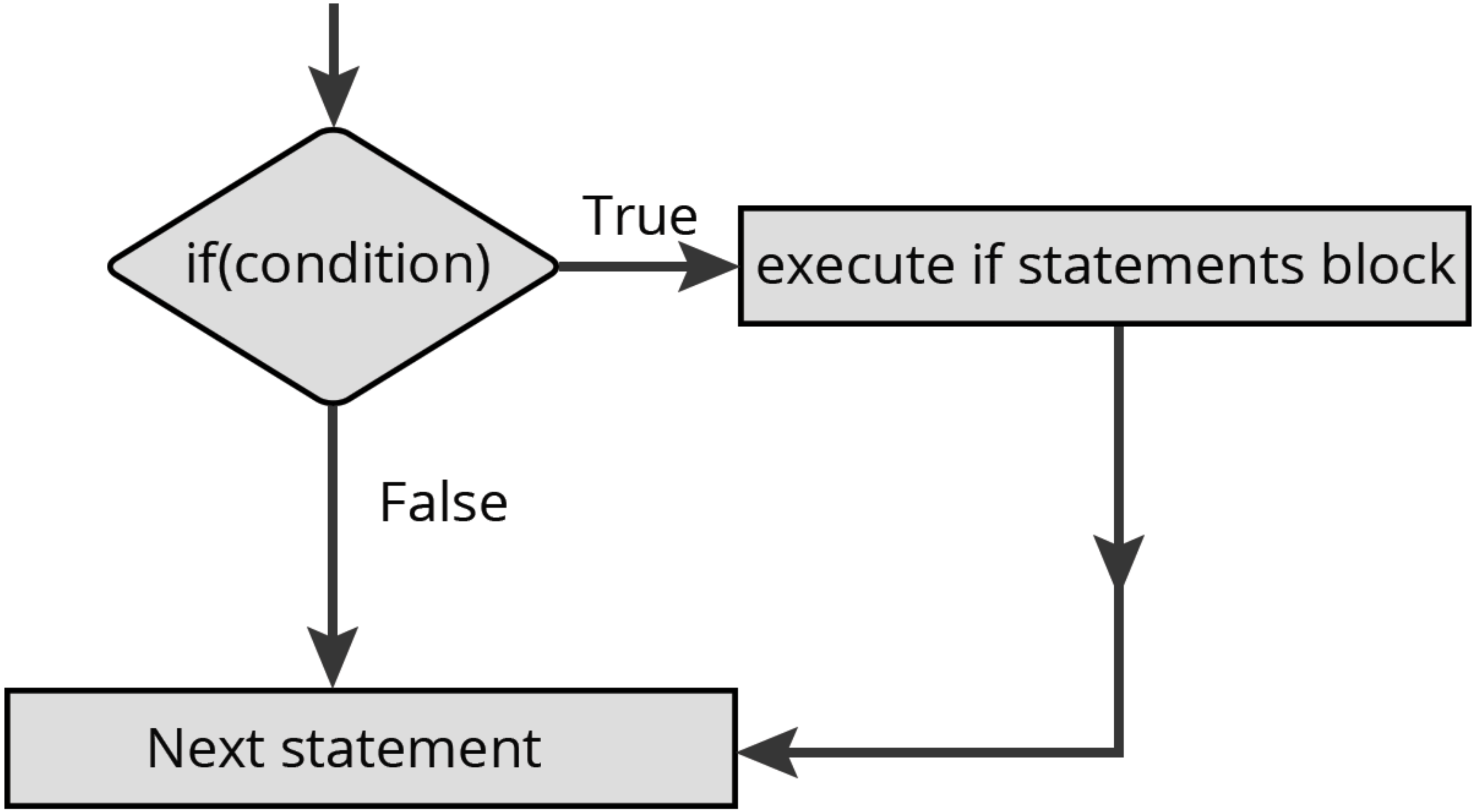
-
  ```
  if(condition)
         statement;
  ```

-
  ```
  if(condition)
  {
         statement$_1$;
         statement$_2$;
         ……
         statement$_n$;
  }
  ```

  Block of statements

- *statement or block of statements* gets executed only if the condition evaluates to *true or non-zero*

Branching: if statement



Example 1

```
int main()
{
  int a;
  printf(“Enter a number”);
  scanf(“%d”, &a);
  if( a > 0 )
  {
     printf(“Number is positive\n”);
  }
  printf(“Rest of the program”);
  return 0;
}
```

O/P:

```
Enter a number
10
Number is positive
Rest of the program
```

Example 2

```
int main()
{
  int a;
  printf(“Enter a number”);
  scanf(“%d”, &a);
  if( a > 0 )
  {
     printf(“Number is positive\n”);
  }
  printf(“Rest of the program”);
  return 0;
}
```

O/P:

```
Enter a number
-5
Rest of the program
```

BITS Pilani
Pilani Campus

Branching: if ladder

Branching: if ladder

A single program can have more than one If statement

```
if (condition_1) {
   //These statements would execute if the condition_1 is true
}
if(condition_2) {
   //These statements would execute if the condition_2 is true
}
.
.
.
if (condition_n) {
   //These statements would execute if the nth condition is true
}
```

Example

```
int a;
printf(“Enter a number”);
scanf(“%d”, &a);

if( a > 0 ) {
   printf(“Number is positive\n”);
   }
if( a < 0 ) {
   printf(“Number is negative\n”);
   }
if( a == 0 ) {
   printf(“You entered zero\n”);
   }
printf(“Rest of the program”);
```

O/P:

```
Enter a number
0
You entered zero
Rest of the program
```

Exercise

Write a C program using only If statements to find the larger value among two numbers

BITS Pilani
Pilani Campus

Branching: if-else statement

Branching: if-else Statement

```
if(condition)
{
   // If- block of statements
}
else
{
   // else-block of statements
}
```

```
if(condition)
   // statement;
else
   // statement;
```

If condition inside the *If* parentheses is true then *If- block of statements is executed, else else-block of statements is executed*

Example

```
int a;
printf(“Enter a number”);
scanf(“%d”, &a);
if( a > 0 )
{
    printf(“Number is positive\n”);
}
else
{
    printf(“Number is either negative or zero\n”);
}
printf(“Rest of the program”);
```

O/P:

```
Enter a number
5
Number is Positive
Rest of the program
```

Example

```
int a;
printf(“Enter a number”);
scanf(“%d”, &a);
if( a > 0 )
{
    printf(“Number is positive\n”);
}
else
{
    printf(“Number is either negative or zero\n”);
}
printf(“Rest of the program”);
```

O/P:

```
Enter a number
-5
Number is either negative or zero
Rest of the program
```

Example 3

```c
#include <stdio.h>
int main() {
   int a = 10, b = 4, c = 10, d = 20;

   if (a > b && c == d)
      printf("a is greater than b AND c is equal to d\n");
   else
      printf("AND condition not satisfied\n");

   if (a > b || c == d)
      printf("a is greater than b OR c is equal to d\n");
   else
      printf("Neither a is greater than b nor c is equal to d\n”);

   if (!a)     printf("a is zero\n");
   else        printf("a is not zero");

   return 0;
}
```

Output:

```
AND condition not satisfied
a is greater than b OR c is equal to d
a is not zero
```

innovate achieve lead

BITS Pilani
Pilani Campus

Branching: if-else ladder

Branching: if-else ladder

```
if (condition$_1$){
   //These statements would execute if the condition$_1$ is true
}
else if(condition$_2$){
   //These statements would execute if the condition$_2$ is true
}
else if (condition$_3$) {
   //These statements would execute if the condition$_3$ is true
}
.
.
else {
   /* These statements would execute none of the previous
   condition is true */
}
```

Example

Write a C program using If-else to find the grade of a student based on following conditions

1. Marks greater than 90 implies grade A
2. Marks between 80 and 90 implies grade B
3. Marks between 70 and 80 implies grade C
4. Else, Grade is Fail

Example: Solution

```
if(Marks > 90)
   printf(“Grade is A”);
else if(Marks <= 90 && Marks >= 80)
   printf(“Grade is B”);
else if(Marks <= 80 && Marks >= 70)
   printf(“Grade is C”);
else
   printf(“Grade is FAIL”);
```

Exercise

Write a code to find whether a character entered is a digit, uppercase alphabet, lowercase alphabet or any other special character

Note:	ASCII value of digits	→ 48 to 57
	ASCII value of A	→ 65
	ASCII value of a	→ 97

innovate achieve lead

BITS Pilani
Pilani Campus

Branching: Nested if-else statement

Branching: Nested if-else statement

If one or more *if* and/or *else* statement is/are present inside the body of another "if" or "else"

```
if (condition1)
{
        if (condition2)
                // statement or block of statements for if
        else
                // statement or block of statements for else
}

else{
        if (condition3)
                // statement or block of statements for if
        else
                // statement or block of statements for else
}
```

Example

```
int a, b, c;
printf(“Enter the numbers”);
scanf(“%d%d%d”, a, b, c);

if(a > b)
{
   if(a > c)
        printf(“A is the largest”);
   else
        printf(“C is the largest”);
}
```

```
else
{
   if(b > c)
        printf(“B is the largest”);
   else
        printf(“C is the largest”);
}
printf(“End of the program”);
return 0;
```

innovate achieve lead

BITS Pilani
Pilani Campus

Control Flow: Ternary Operator

Ternary Operator

- An alternative way to write if…else construct:

```
if (expr1)
        expr2 ;
else
        expr3 ;
```

- **Syntax:** `expr1? expr2:expr3`

- Only one of **`expr2`** and **`expr3`** is evaluated
- If **`expr2`** and **`expr3`** are of different types, the type of the result is determined by the type conversion rules.

Example

```
A = 43, B = 7, C = 0, D = 0
```

```
A + B == 50 ?  C = 10 : C = 15
```

- O/P: C = 10

```
A > C ? printf("Hello"): printf("World");
```

- O/P: ?

```
A < C ? printf("Hello"): printf("World");
```

- O/P: ?

Exercise

Write a statement (using ternary operator) to find out the largest of the three integers `a`, `b` and `c`, and store the value in `max`.

Control Flow: Switch Statement

Switch Statement

- A multi-way decision that tests whether an expression matches one of a number of constant integer values, and branches accordingly.

Syntax:

```
switch (expr) {
        case const-expr :   statements
                            break;
        case const-expr :   statements
                            break;
        default:            statements
                            break;
}
```

Example 1

```
#include <stdio.h>
int main() {
  int language = 10;
  switch (language) {
     case 1: printf("C#\n");
           break;
     case 2: printf("C\n");
           break;
     case 3: printf("C++\n");
           break;
     default:
           printf("Other programming language\n");
  }
}
```

What is the output?

Example 2

```
#include <stdio.h>
int main() {
  int number=5;
  switch (number) {
      case 1:
      case 2:
      case 3:
            printf("One, Two, or Three.\n");
            break;
      case 4:
      case 5:
      case 6:
            printf("Four, Five, or Six.\n");
            break;
      default:
            printf("Greater than Six.\n");
}}
```

What is the output?

innovate achieve lead

BITS Pilani
Pilani Campus

Thank you

Q & A