United International University

Dept. of Electrical and Electronic Engineering (EEE)

Course No. : **EEE 122**
Course Title: **Structured Programming Laboratory**

Lab Sheet 3
Selection Statements in C

Outcomes

After finishing this lab students should be able to ...

1. write simple decision making statements.
2. use the if selection statement and the **`if...else`** selection statement to select actions.
3. use **`switch`** Statement understand multiple selection using the switch selection statement.

Contents

1 C - Decision Making

Decision making structures require that the programmer specifies one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

C programming language assumes any **non-zero** and **non-null** values as **`true`**, and if it is either **zero** or **null**, then it is assumed as **`false`** value.

C programming language provides the following types of decision making statements.

1.1 `if` statement

An **`if`** statement consists of a Boolean expression followed by one or more statements.

Syntax

```
if(condition) {
    // statement(s)
}
```

Flow Diagram

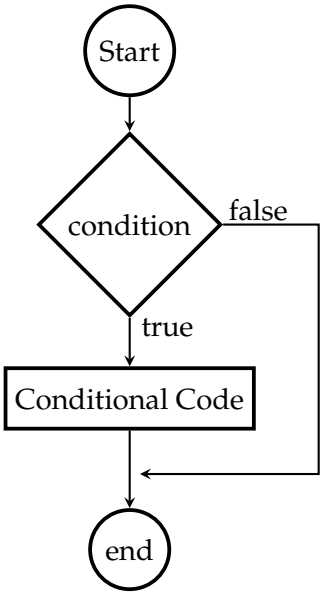

Fig: **`if`** condition flow chart

Example

```
#include <stdio.h>

int main (void) {
    int a;
    printf("\nEnter a value  ");
    scanf("%d",&a);
    if( a < 20 ) {
        /* if condition is true then print the following */
        printf("a is less than 20\n" );
    }
    printf("value of a is : %d\n", a);
    return 0;
}
```

1.2 `if...else` statement

An **`if`** statement can be followed by an optional **`else`** statement, which executes when the Boolean expression is false.

Syntax

```
if(condition) {
// statement(s)
}
else {
// statement(s)
}
```

Flow Diagram

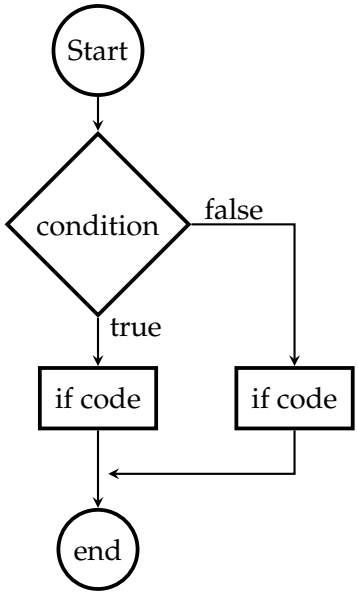

Fig: **`if-else`** condition flow chart

Example

```c
#include <stdio.h>

int main (void) {
    int a;
    printf("\nEnter a value  ");
    scanf("%d",&a);
    if( a < 20 ) {
        /* if condition is true then print the following */
        printf("a is less than 20\n" );
    }
    else{
        /* if condition is false then print the following */
        printf("a is greater than 20\n" );
    }
    printf("value of a is : %d\n", a);
    return 0;
}
```

1.3 `If...else if...else` statement

An **if** statement can be followed by an optional **else if...else** statement, which is very useful to test various conditions using single **if...else if** statement. Once an **else if** succeeds, none of the remaining **else if**'s or **else**'s will be tested.

Syntax

```c
if(condition 1) {
/* Executes when the condition 1 is true */
}
else if(condition 2) {
/* Executes when condition 2 is true */
}
else if(condition 3) {
/* Executes when the condition 3 is true */
}
...
...
...
else  {
/* executes when the none of the above condition is true */
}
```

Flow Diagram

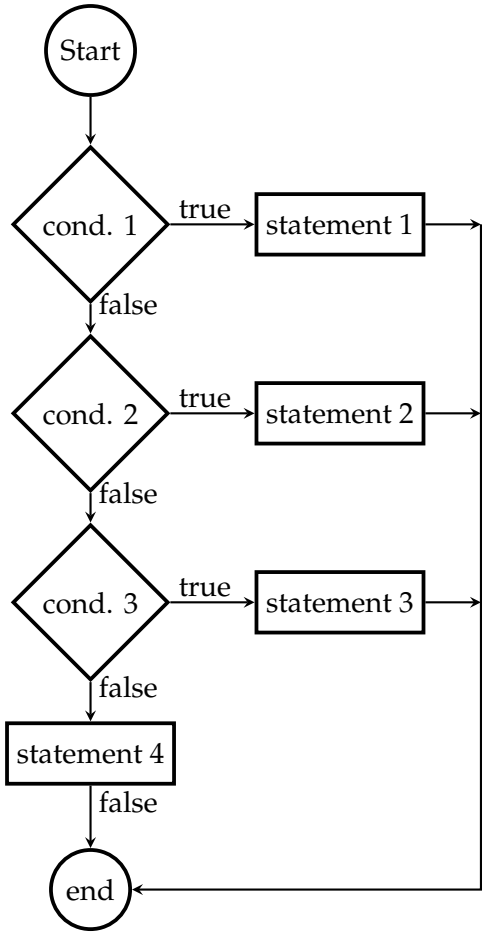

Fig: **`if-else`** condition flow chart

Example

```c
#include <stdio.h>

int main (void) {
    int a;
    printf("\nEnter a value  ");
    scanf("%d",&a);
    if( a < 20 ) {
        /* if condition 1 is true then print the following */
        printf("a is less than 20\n" );
    }
    else if( a < 30 ) {
        /* if condition 2 is true then print the following */
        printf("a is less than 30\n" );
    }
    else if( a < 40 ) {
        /* if condition 3 is true then print the following */
        printf("a is less than 40\n" );
    }
    else{
        /* if all condition is false then print the following */
        printf("a is greater than 40\n" );
    }
    printf("value of a is : %d\n", a);
    return 0;
}
```

1.4 C - nested `if` statements

It is always legal in C programming to **nest if-else** statements, which means you can use one if or else if statement inside another if or else if statement(s).

Syntax

```
if(condition1) {
/* Executes when condtion1 is true */
    if(condition2) {
        /* Executes when condition2 is true */
    }
}
```

Example

```
#include <stdio.h>
int main(void) {
    /* local variable definition */
    int a = 100;
    int b = 200;
    /* check the boolean condition */
    if( a == 100 ) {
        /* if condition is true then check the following */
        if( b == 200 ) {
            /* if condition is true then print the following */
            printf("Value of a is 100 and b is 200\n" );
        }
    }
    printf("Exact value of a is : %d\n", a );
    printf("Exact value of b is : %d\n", b );
    return 0;
}
```

1.5 C -`switch` statements

A **`switch`** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each **`switch`** case.

Syntax

```
switch(expression) {
    case constant-expression  :
                statement(s);
                break; /* optional */

    case constant-expression  :
                statement(s);
                break; /* optional */

            /* you can have any number of case statements */

    default : /* optional */
                statement(s);
}
```

Flow Diagram

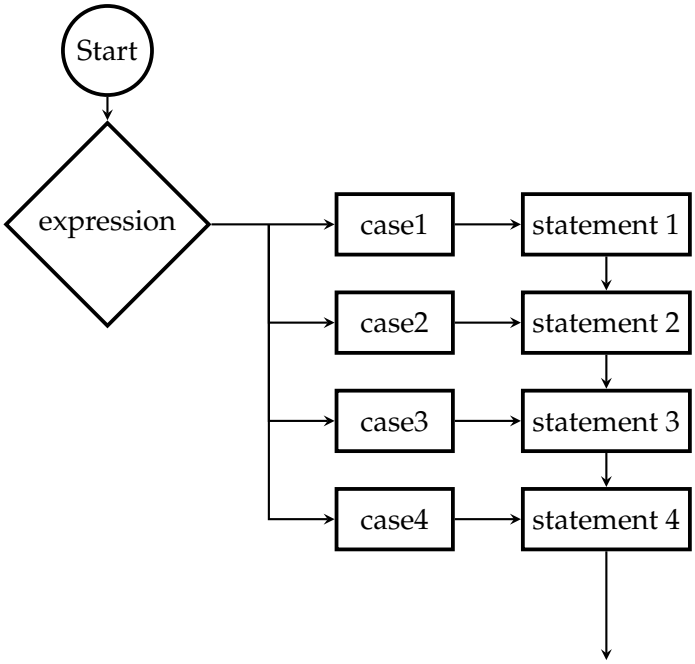

Fig: **`switch`** statement flow chart

Example

```c
#include <stdio.h>

int main (void) {
    char grade = 'B';

    switch(grade) {
        case 'A' :
            printf("Excellent!\n" );
            break;
        case 'B' :
        case 'C' :
            printf("Well done\n" );
            break;
        case 'D' :
            printf("You passed\n" );
            break;
        case 'F' :
            printf("Better try again\n" );
            break;
        default :
            printf("Invalid grade\n" );
    }

    printf("Your grade is  %c\n", grade );
    return 0;
}
```

2 Programming Examples

Example: 1

Description: Write a C program that check the condition in if. If it is true, it's print a message

Source Code

```
#include <stdio.h>
int main(void){
    int a=5;;
    if(a==5)
    printf("The entered number is %d.\n",a);
    return 0;
}
```

Output

```
The entered number is 5
```

Example: 2

Description: Write a C program that finds and displays maximum of two numbers.

Source Code

```
#include <stdio.h>
int main(void){
    int a,b;
    printf ("Enter the first number:");
    scanf("%d",&a) ;
    printf ("Enter the second number:");
    scanf ("%d",&b) ;
    if (a>b)
        printf("%d is larger than %d",a,b);
    else
        printf("%d is larger than %d",b,a);
    return 0;
}
```

Output

```
Enter the first number:10
Enter the second number:20
20 is larger than 10
```

Example: 3

Description: Write a C program that finds and displays maximum of three numbers.

Source Code

```
#include <stdio.h>
int main(void){
    int a,b,c,large;
    printf ("Enter three numbers ") ;
    scanf("%d%d%d",&a,&b,&c) ;
    if(a>b){
        if(a>c)
            large=a;
        else
            large=c;
    }
    else{
        if(b>c)
            large=b;
        else
            large=c;
    }
    printf("Largest number is %d\n", large) ;
    return 0;
}
```

Output

```
Enter three numbers 10 20 8
Largest number is 20
```

Example: 4

Description: Write a C program to print equivalent letter grade from numeric value. Compute the grades as given in the following table.

Marks(n)	Grade
n >=85	A
70 <= n <85	B
55 <= n <70	C
40 <= n <54	D
n <40	F

Source Code

```
#include <stdio.h>
int main(void){
    float num;
    char grade;
    printf("Enter marks of the subject: ");
    scanf("%f",&num) ;
    if(num>=85)
        grade= 'A' ;
    else if(num>=70)
        grade='B';
    else if (num>=55)
        grade= 'C' ;
    else if (num>=40)
        grade= 'D' ;
    else
        grade='F';
    printf ("Mark is %f\nGrade is %c\n", num,grade);
    return 0;
}
```

Output

```
Enter marks of the subject: 45
Mark is 45.000000
Grade is D
```

Example: 5

Description: Write a C program to perform arithmetic calculation on integers using switch operator.

Source Code

```
#include <stdio.h>
int main(void){
    char op;
    int a,b;
    printf ("Enter number operator and anothernumber: ");
    scanf("%d %c %d", &a, &op, &b) ;
    switch(op){
        case '+':
            printf ("Sum=%d\n", a+b) ;
            break;
        case '-':
            printf ("Difference = %d\n", a-b);
            break;
        case '*':
            printf("Product= %d\n",a*b);
            break;
        case '/':
            printf("Quotient= %d\n",a/b);
            break;
        case '%':
            printf("Quotient= %d\n",a%b);
        default:
            printf("\nInvalid choice.");
}
return 0;
}
```

Output

Run1:

```
Enter number operator and another
number:
100+50
Sum=150
```

Run2:

```
Enter number operator and another
number:
100/50
Quotient= 2
```

3 Practice session

Sl	Source Code
Practice 1	

```c
#include <stdio.h>
int main(void){
    int a=1,b=3;
    if(a==2)
        if(b==3)
            printf("Prince");
printf("Queen");
return 0;
}
```

Practice 2	

```c
#include <stdio.h>
int main(void){
    int a=1,b=3;
    if(a==2)
        if(b==3)
            printf("Prince");
        else
            printf("Queen");
    else
        printf("King");
    return 0;
}
```

Practice 3	

```c
#include <stdio.h>
int main(void){
    int temp=32;
    if ( temp < 80 )
        if ( temp > 60 )
            printf("Nice day!");
        else
            printf("Sure IT is hot!");
    return 0;
}
```

Practice 4	

```c
#include <stdio.h>
int main(void){
int a=2,x=10;
if(a==2)
    if (x==8)
        printf("a is equal to 2 and x is equal to 8 n");
    else
        printf("a is not equal to 2 ");
return 0;
}
```

4 Lab Assignments

1. An Electric power distribution company charges its domestic consumers as follows :

Consumption units	Rate of charge
0-100	4.5 taka per unit
101-200	100 taka plus 5 taka per unit excess of 100
201-400	250 taka plus 6 taka per unit excess of 200
401 and above	400 taka plus 7 taka per unit excess of 400

   Write a C program which reads the customers amount of power consumed and prints the amount to be paid by the customer.

2. Write a program in C which gives the solution of a quadratic equation ,using the formula $x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$. You must find any kind of roots (either real or maginary).In case of imaginary roots ,the roots should be of the form c+id and c- id. Keep the option that when one enter a = 0 as input, it prints: This is not a quadratic equation.

3. Write a program in C asking the user to enter 2 digit number, then prints the English word for it. Suppose you enter '41' the printf function prints out 'forty one'. Use switch statement for this purpose.

Acknowledgment

First OBE version, prepared by:
B.K.M. Mizanur Rahman,
Assistant Professor,
Department of EEE, UIU

Second Update , prepared by:
Nazmul Alam,
Part Time Faculty,
Department of EEE, UIU