Rajarata University of Sri Lanka

Faculty of Applied Sciences

COM 1407 – Computer Programming

Practical – Flow Control Structures (Part 1) - Decision Making

---

Learning Outcomes:

After successfully completing the practical, students will be able to:

- Be familiar with the Decision making in C.
- Get knowledge about the general structure of the if, if..else, if..elseif..else and nested if , switch statements.
- Use relevant decision making constructs in C programs.

---

if statement

- An **if statement** consists of a Boolean expression followed by one or more statements
- Syntax

```
if(boolean_expression)
{
        Statements will execute if the Boolean expression is true
}
```

if..else statement

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

```
if(boolean_expression)
 {
        Statements will execute if the Boolean expression is true
 }
else
{
        Statements will execute if the Boolean expression is false
}
```

if..elseif..else

- 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

- Syntax

```
if(boolean_expression)
 {
        Statements will execute if the Boolean expression is true
 }
elseif(boolean_expression 2)
{
        Statements will execute if the Boolean expression 2 is true
}
elseif(boolean_expression 3)
{
        Statements will execute if the Boolean expression 3 is true
}
else
{
        Statements will execute if none of the above conditions is true
}
```

Nested if statements

- Means that you can use one **if or else if statement inside another if or else if statement(s).**

- Syntax

```
if( boolean_expression 1)
{

  Executes when the Boolean expression 1 is true
        if(boolean_expression 2)
        {
                Executes when the Boolean expression 2 is true

        }
}
```

- In if statements you can use logical operators as the condition.
  - Logical AND Operator = **&&**
  - Logical OR Operator = **||**
  - Logical NOT Operator = **!**

- Syntax

```
if(a==100 && b==300)
{
        printf("wrong");
}
else
{
        printf("right");
}
```

Conditional / Ternary Operator (?:)

- The conditional operator is kind of similar to the if-else statement as it does follow the same algorithm as of if-else statement but the conditional operator takes less space.

- Syntax

variable = Expression1 ? Expression2 : Expression3

Here, **Expression1** is the condition to be evaluated. If the condition (**Expression1**) is True then **Expression2** will be executed and the result will be returned.
Otherwise, if the condition (**Expression1**) is false then **Expression3** will be executed and the result will be returned.

For better understanding, try the below code

```
#include <stdio.h>

int main()
{
   int n1 = 5, n2 = 10, max;

   max = (n1 > n2) ? n1 : n2;

   printf("Largest number between %d and %d is %d. ", n1, n2, max);

  return 0;
}
```

Exercises

1. Write and run a program that reads the user's age and then prints
    I. You are a Child. - If the age<18
    II. You are a Senior citizen-If age>=18

2. Write and run a program that reads the user's age and then prints
    I. You are a Child. - If the age<18
    II. You are an Adult-If18<=age<65
    III. You are a Senior citizen- If age>=65

3. The marks (M1, M2, M3) obtained by a student in 3 different subjects are input through the keyboard. The students are categorized as per the following rules.
    I. Average above or equal to 75-A
    II. Average between 55 and 74 – B
    III. Average between 40and 54-C
    IV. Average below 40-F

4. Write and run a program that simulates a simple calculator. It reads two integers and a character.
    I. If the character is a +, the sum is printed;
    II. If it is a -, the difference is printed;
    III. If it is a *, the product is printed;
    IV. If it is a /, the quotient is printed
    V. If it is a %, the remainder is printed

5. A company insures its drivers in the following cases
    I. If the driver is married
    II. If the driver is unmarried male and above 30 years of age
    III. If the driver is unmarried, female and above 25 years of age

    Read age, sex and marital status through the keyboard and output whether the driver is insured or not

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 (variable)
{
        case value1 :
        statement 1; //If the value of Variable equals to Value1, This code will execute
        break;

        case value2 :
        statement 2; //If the value of Variable equals to Value1, This code will execute
        break;

        case value3 :
        statement 3; //If the value of Variable equals to Value1, This code will execute
        break;

        default:
        statement default; //If the value is equal to none of the above, This code will execute
}
```

When we consider switch statements following rules apply

- The **constant-expression** for a case must be the same data type as the variable in the switch.

- When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.

- When a **break** statement is reached, the **switch** terminates, and the flow of control jumps to the next line following the switch statement.

- Not every case needs to contain a break. If no break appears, the flow of control will fall through to subsequent cases until a break is reached.

For better understanding the switch statement try below code and see the output.

```
#include<stdio.h>
int main()
{
char Grade = 'B';

switch (Grade)

{
        case 'A' : printf("Grade is A\n");
        break;
        case 'B' : printf("Grade is B\n");
        break;
        case 'C' : printf("Grade is C\n");
        break;
        case 'D' : printf("Grade is D\n");
        break;
        default : printf("none of the above\n");

}
```

Exercises

1. Write a program to get a number from the user (numbers from 0 to 10) and print the entered number.
2. Write a program to display the month, according to the given month number (keyboard input). Example: If you input 1, output should be January.
3. Write a program to check whether user input alphabet is a vowel (a,e,i,o,u) or not.
4. Write a program to Bank transaction process. The user inputs the transaction code and the transaction amount. You should display the final balance.

Transaction codes	**Balance**
Withdrawal - 'W'	Balance= balance - amount
Deposit - 'D'	Balance= balance + amount