C# and .NET Framework – Lab sheet 02

Lab 02 – Conditional

1. Boolean Expressions

A boolean expression have only two possible values: *true* and *false*. In C# , boolean expressions can be written using one or more of the following symbols:

Relation	C# notation	Example	Meaning
=	==	x == y	x is equal to y
/=	!=	x != y	x is not equal to y
>	>	x > y	x is greater than y
≥	>=	x >= y	x is greater than or equal to y
<	<	x < y	x is less than y
≤	<=	x <= y	x is less than or equal to y

Boolean expressions can be combined or modified to form a more complex expression using one of the following operators.

- && combines two boolean expressions using the operator **AND**. For instance, the expres-sion (x>10) &&(x<10) is true when *x* is between 1 and 10.
- || combines two boolean expressions using the operator **OR**. For instance, the expression (x<1) ||(x>10) is true when *x* is less than 1 or greater than 10.
- ! negates the truth value of a boolean expression. For example, !(x==1) is true when *x* is not equal to 1.

Lesson 1.1: Let *x*, *y* and *z* be of type int **and *ch* of type** char**. Describe the condition that makes each of the following boolean expressions true**

Expression	Condition to be true
x > 2	true when *x* is greater than 2
x%2 == 0	true when *x* is an even number
(x%5 == 0)	**True when x is multiple of 5**
(x%y == 0)	**True when x is multiple of y**
((x%y == 0) && (z%y == 0))	**True when x & z both are multiples of y**
ch == ’a’	**True when ch is ‘a’**
((ch >= ’a’) && (ch <= ’z’))	**True when ch is lower case alphabet**
((ch >= ’A’) && (ch <= ’Z’))	**True when ch is upper case alphabet**
((ch >= ’0’) && (ch <= ’9’))	true if *ch* is a character between ‘0’ and ’9’
(ch != ’*’)	**True when ch is not asterisk**
!(ch == ’*’)	**True when ch is not asterisk**

2. if and if...else Statements

if statement is a conditional statement that controls whether a specified statement should be executed, based on the given *condition*. There are two forms of usage, as follows.

- **Form 1: if statement** the statement will be executed when the condition is true

```
if  ( condition )                {}
      statem ent;                 // execute  if  condition  ==  true
```

In C# , a pair of braces ( ) are used to group multiple statements together, which is useful when we need more than one statement to be executed when the condition is true.

```
if ( condition ) {
      statem ent1 ;      // execute      if    condition       == true

}
      statem ent2 ;      // execute      if    condition       ==  true
      statem ent3 ;      // execute      if    condition       ==  true
```

Example 2.1: Consider the following pseudo-code

```
if student's score is greater than or equal to 60 Print "Passed"
otherwise
    Print "Failed"
```

which means "if the student's score is greater than or equal to 60, show *Passed* ; otherwise show *Failed*." Using this pseudo-code, we can write code in C# as follows:

```
if (score >= 60)
Console.WriteLine("Passed");
if (score < 60)
   Console.WriteLine("Failed");
```

Lesson 2.1 **Type the following program into your editor, then answer the questions**

```
using System ;
class  Lab 321 C  {
    static  void  Main () {
          int  N ;
          N = int . Parse ( Console . Read Line ());
          if ( N < 0)
              Console . W rite Line (" N egative . N um ber");
          else
              Console . W rite Line (" Positive . N um ber ");
}
```

- Give three different values for the variable *N* to make the program display "Negative Number".
    - **-1**
    - **-12**
    - **-11**
- Give three different values for the variable *N* to make the program display "Positive Number"
    - **1**
    - **45**
    - **76**
- If the user enters 0 to the program, what will be the result?
    - **If 0 is entered, it goes to else condition. 0<0 is false.**

- Modify the program so that it can also display "Zero Number" (in addition to "Positive Number" and "Negative Number") if the user enters 0. Write the program

```
static void Main(string[] args)
{
    int N = int.Parse(Console.ReadLine());

    if (N < 0)
        Console.WriteLine("Negative . Number");
    else if (N == 0)
        Console.WriteLine("Zero . Number");
    else
        Console.WriteLine("Positive . Number ");
}
```

3. Quadrant indicator

The following incomplete C# program attempts to identify the quadrant of the input (x, y) coordinates. If the input coordinates happen to be on either X-axis or Y-axis, the program will display “I don’t know.”

```
using System ;
class Quadrant {
        static   void     M ain ()
        {
                Console . W rite (" Enter . X:.");
                int x = int . Parse ( Console . ReadLine ()); Console . W rite (" Enter .Y: ." );
                int y = int . Parse ( Console . ReadLine ());
                if        (____( a)____)
                        Console . W rite Line (" ({0} ,{1}) . is. in . Q1 .",x, y);
                if        (____   ( b)____)
                        Console . W rite Line (" ({0} ,{1}) . is. in . Q2 .",x,  y);
                if        (____   ( c)____)
                        Console . W rite Line (" ({0} ,{1}) . is. in . Q3 .",x,  y);
                if        (____   ( d)____)
                        Console . W rite Line (" ({0} ,{1}) . is. in . Q4 .", x, y);
                if        (____   ( e)____)
                        Console . W rite Line (" I. don ’t. know .");
        }
}
```

Complete the program above by determining what should be put in the blanks marked (a)-(e).

Blank	Boolean expression
(a)	x > 0 && y > 0
(b)	**x < 0 && y > 0**
(c)	**x < 0 && y < 0**
(d)	**x > 0 && y < 0**
(e)	**X==0 ||y==0**

4. Body Mass Index (BMI)

The BMI exercise done in class did not give the complete classification. The complete list is shown below:

BMI	Interpretation
BMI < 18.5	Underweight
18.5 ≤ *BMI* < 25	Normal
25≤*BMI* <30	Overweight
BMI ≥ 30	Obese

Complete the following BMI calculator program by filling in appropriate boolean expressions in the provided blanks.

```
using System ;
class  BM ICalc  {
      static      void      M ain () {
            Console . Write (" Enter . your . w eight .( in. kg ):."); double w = double .
            Parse ( Console . ReadLine ()); Console . Write (" Enter . your . height . ( in .
            m ):."); double h = double . Parse ( Console . ReadLine ()); double bmi = w /(
            h* h);

            Console . W rite Line (" Your . BMI . is . {0: f2 }.",  bmi );

            if (    _____( a)  _____  )
                  Console . W rite Line (" You . are . unde rweight .");
            else     if    (____( b)  ____   )
                  Console . W rite Line (" You . are . normal . ");
            else    if    ( _____( c)  ____   )
                  Console . W rite Line (" You . are . overw eight ."); else

                  Console . W rite Line (" You . are . obese .");
      }
}
```

Blank	Boolean expression
(a)	**bmi<=18.5**
(b)	**Bmi>=18.5 && bmi<25**
(c)	**Bmi>=25 && bmi<30**

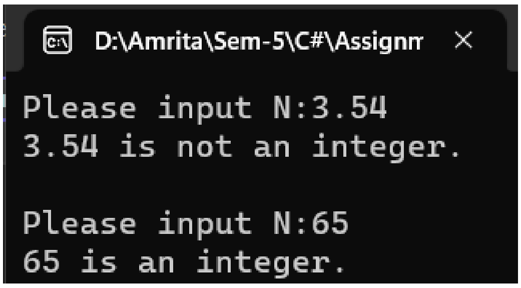
5. Programming Exercises

Write a C# program to determine whether the input number is an integer. (Hint: Use the method Math.Round())

```
static void Main()
 {
     Console.Write("Please input N:");
     double num = double.Parse(Console.ReadLine());

     if (num - Math.Round(num) == 0)
         Console.WriteLine($"{num} is an integer.");
     else
         Console.WriteLine($"{num} is not an integer.");

     Console.ReadLine();
 }
```



```
Please input N:3.54
3.54 is not an integer.

Please input N:65
65 is an integer.
```