# ESC101 : Fundamental of computing

**Quiz 1(B) Solution  Marks=10  25 September, 2008**

*In case some tutor made any announcement contrary to what is mentioned in quiz paper, he has to grade the quiz of his section according to his announcement.*

**Question 1** (each blank carries 0.5 marks) Fill in the blanks the method `IsPrime` whose parameter is a positive integer of type `int`, and it returns true if the number is prime and returns false if the number is not prime.

```
public static boolean IsPrime(int n)
   {   int t = 2 ;
       boolean flag = true;

       while(t*t <= n &&  flag==true)
       {   if(n%t==0) flag = false;
           t = t+1;
       }
       if(n==1 || flag==false) return false;
       else return true;
   }
```

**Question 2.** We want to create a class Box. The attributes of a box are its length, breadth, and width. It should have two constructors :
`Box(double x)` : to construct a Box with its length, breadth and width equal to `x`.
`Box(double x, double y, double z)` : to construct a Box with length `x`, breadth `y`, and width `z`.
**Note :** you may assume that the constructor is called with appropriate arguments such that the resulting box has length greater than or equal to both its breadth as well as its width.
You have to design a method `Volume()` which returns volume of the current box. You also have to design a method `CanEnclose(Box B)` which returns true if the current Box can enclose the Box B completely. Please fill in the blanks the following description of Box class. **You have to ensure that once a Box is created it should not be possible to change its length, breadth and width.**

```
public class Box
{
    private  double length;

    private  double breadth;

    private double width;

    public Box(double x)     {length=x; breadth=x; width=x;}

    public Box(double x, double y, double z)
    {length=x; breadth=y; width=z;}

    public double Volume()      {return (length*breadth*width)}

    public boolean CanEnclose(Box B)
    {
       if(length > B.length && (breadth > B.breadth && width > B.width ||
                              breadth > B.width && width > B.breadth)))
           return true;
       else return false;
    }
}
```

**Marking scheme :**The method CanEnclose() carries 2 marks, and each of the remaining blanks has 0.5 mark each.
It is fine if you use >= instead of > in the method CanEnclose() above. But you will be given just one mark if you wrote either

```
(length > B.length &&  breadth > B.breadth && width > B.breadth)
or
(length >= B.length && breadth >= B.breadth && width >= B.breadth)
```