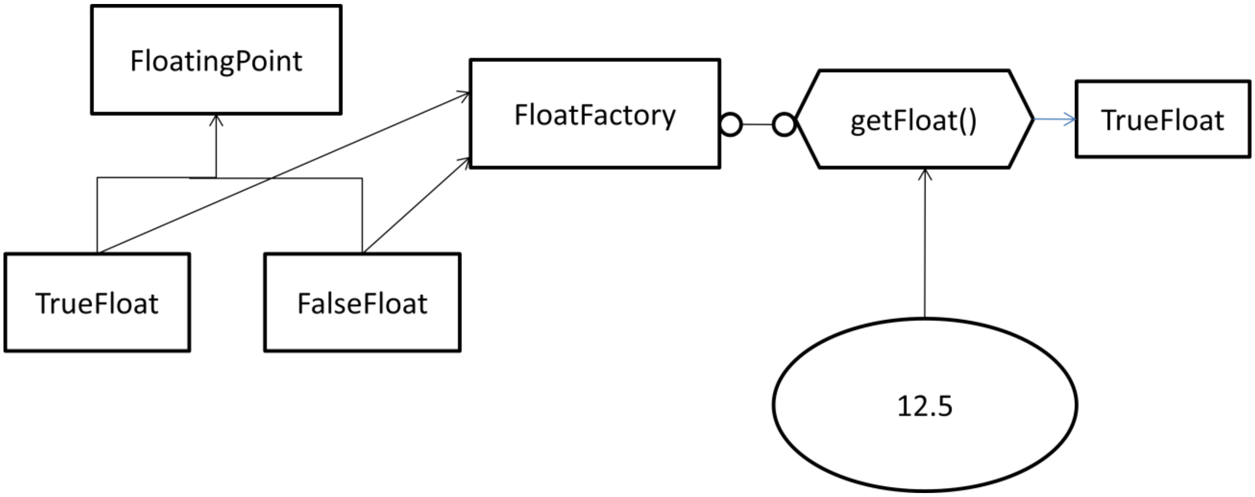
# Bangladesh University of Engineering and Technology (BUET)

# CSE 202 (Object Oriented Programming Language Sessional)

# Online on Java

A Factory pattern is one that returns an instance of one of several possible classes depending on the data provided to it.



**Input:**

You have to take a **String** as input from console (e.g. 12.5). The you have to convert it in mantissa and exponent form.

$$1.2345 = \underbrace{12345}_{\text{mantissa}}\times\overbrace{10^{-4}}^{\text{exponent}}$$

For 12.5, **mantissa** = 125 and **exponentfrequency** = -1

**Output**:

Output will be the **Rounded** form of the **input**. You cannot use **floor** or **ceil** functions for rounding.

**Sample Input and Output:**

1.

```
100.234
Mantissa: 100234 Exponential Frequency: -3
Rounded Value is: 100
```

2.

```
100.56345
```

```
       Mantissa: 10056345 Exponential Frequency: -5
       Rounded Value is: 101
```

**Sample Code:**

```
import java.util.Scanner;

class FloatPoint
{
    Protected int mantissa,exponentFreq;
}
 class TrueFloat extends FloatPoint
 {
    public TrueFloat(String s)
    {
       // logic to obtain mantissa and exponentFreq
    }
 }
class FalseFloat extends FloatPoint
{
  public FalseFloat(String s)
  {
       // logic to obtain mantissa and exponentFreq
  }
}

class FloatFactory
{
       public FloatPoint getFloat(String entry)
       {
              //determine which class to return
       }
}
public class Main {
   private static void roundNum(FloatPoint fp)
   {
       System.out.println("Rounded Value is: "+ value);
   }

  public static void main(String[] args) {
    // TODO code application logic here
    Scanner sc = new Scanner(System.in);
    String s = sc.next();
    FloatFactory ff = new FloatFactory();
    FloatPoint fp= ff.getFloat(s);
    System.out.println("Mantissa: "+ fp.mantissa+" Exponential Frequency: "+ fp.exponentFreq);
    roundNum(fp);
  }
}
```