JAVA ASSIGNMENT 1

Ques 1: Write a method that overloads the ‘speak’ method by taking in a name and printing “Hello” with that name.

Sol: Main.java

```
class Name{
   String speak()
   {
      return ("Initial method");
   }
   String speak(String name)
   {
      return name;
   }
}


public class Main{
   public static void main(String args[])
   {
      Name nm = new Name();
      System.out.println("Hello " + nm.speak("Puja") );
   }
}
```

Output:

Hello Puja

Ques1:

Main.java

```
1  class Name{
2      String speak()
3      {
4          return ("Initial method");
5
6      }
7      String speak(String name)
8      {
9          return name;
10     }
11 }
12
13 public class Main{
14     public static void main(String args[])
15     {
16         Name nm = new Name();
17         System.out.println("Hello " + nm.speak("Puja") );
18     }
19 }
```

input

```
Hello Puja
```

Ques 2: Override the taste method from the Candy class in the Chocolate class to return “tastes chocolately”. It should print “tastes sweet!” and then “tastes chocolately”.

Sol: Main.java

```
class Candy{
   public void tastes()
   {
      System.out.println("tastes sweet!");
   }
}
class Chocolate extends Candy{
   public void tastes()
   {
      super.tastes();
      System.out.println("tastes chocolately");
   }
}

public class Main{
   public static void main(String args[]){
      Chocolate c = new Chocolate();
      c.tastes();
   }
}
```

Output:

```
tastes sweet!
tastes chocolately
```

Ques2:

Main.java

```java
class Candy{
    public void tastes()
    {
        System.out.println("tastes sweet!");
    }
}
class Chocolate extends Candy{
    public void tastes()
    {
        super.tastes();
        System.out.println("tastes chocolately");
    }
}

public class Main{
    public static void main(String args[]){
        Chocolate c = new Chocolate();
        c.tastes();
    }
}
```

input

```
tastes sweet!
tastes chocolately
```

Ques 3 : Write a java program to print the given string in the format as in the output.

Input : text@text,text!text$text&text

Output:

Text

Text

Text

Text

Text

Text

Sol:

```
public class Main {
	public static void main(String args[]) {
		String str = "$#*text@text,text!text$$text,&!text*&%^";
	str = str.replaceAll("[^a-zA-Z]"," "); //replaces everything except alphabets.
	String p ="";
		StringBuffer s = new StringBuffer();   //mutable string
		char ch = ' ';                    //only for first character capitalized
		for (int i = 0; i < str.length(); i++) {
			if (ch == ' ' && str.charAt(i) != ' ')
				s.append(Character.toUpperCase(str.charAt(i)));
			else
				s.append(str.charAt(i));
			ch = str.charAt(i);
		}
	p = s.toString().trim();                //stringbuffer to string
	p = p.replaceAll("\\s+"," ").trim();       //removing extraspaces of repeated non-chars
	for(int i = 0 ; i < p.length(); i++ )      //printing in next line
```

```
        {
            if(p.charAt(i)!=' ')
            {
                System.out.print(p.charAt(i));
            }
            else
            {
                System.out.println("");
            }}}}
```

Output:

Text

Text

Text

Text

Text

Text

Main.java

```java
public class Main {
    public static void main(String args[]) {
        String str = "$#*text@text,text!text$$text,&!text*&%^";
        str = str.replaceAll("[^a-zA-Z]"," "); //replaces everything except alphabets.
        String p ="";
        StringBuffer s = new StringBuffer();   //mutable string
        char ch = ' ';                         //only for first character capitalized
        for (int i = 0; i < str.length(); i++) {
            if (ch == ' ' && str.charAt(i) != ' ')
                s.append(Character.toUpperCase(str.charAt(i)));
            else
                s.append(str.charAt(i));
            ch = str.charAt(i);
        }
    p = s.toString().trim();                   //stringbuffer to string
    p = p.replaceAll("\\s+"," ").trim();       //removing extraspaces of repeated non-chars
    for(int i = 0 ; i < p.length(); i++ )      //printing in next line
    {
        if(p.charAt(i)!=' ')
        {
            System.out.print(p.charAt(i));
        }
        else
        {
            System.out.println("");
        }}}}
```

QUES-3: PROGRAM AND OUTPUT

Main.java

```java
public class Main {
    public static void main(String args[]) {
        String str = "$#*text@text,text!text$$text,&!text*&%^";
        str = str.replaceAll("[^a-zA-Z]"," "); //replaces everything except alphabets.
        String p ="";
        StringBuffer s = new StringBuffer();   //mutable string
        char ch = ' ';                         //only for first character capitalized
        for (int i = 0; i < str.length(); i++) {
            if (ch == ' ' && str.charAt(i) != ' ')
                s.append(Character.toUpperCase(str.charAt(i)));
            else
                s.append(str.charAt(i));
            ch = str.charAt(i);
        }
    p = s.toString().trim();                   //stringbuffer to string
    p = p.replaceAll("\\s+"," ").trim();       //removing extraspaces of repeated non-chars
    for(int i = 0 ; i < p.length(); i++ )      //printing in next line
```

input

```
Text
Text
Text
Text
Text
Text

...Program finished with exit code 0
Press ENTER to exit console.
```

Ques 4: Write a program to dynamically initialize memory for storing following values in the following locations:-

2 4 6 8

3 6 9

4 8

5

SOL:

```
class dynamicArray {
        int darr[],count;
        public dynamicArray(int length)
        {    darr = new int[length]; }
         public void displayshape()
   {
       int k = count/2;
         for( int i=0 ; i<=k ;i++)
         {
            for(int j = 0; j<count; j++ )
            {
               if(darr[j]%darr[i] == 0  )
                   System.out.print(darr[j]+" ");
            }
            System.out.println();
         }
   }
```

```
    public void insert(int value)
    {
        if (darr.length == count) {
            int newArr[] = new int[2 * count];
            for (int i = 0; i < count; i++) {
                newArr[i] = darr[i];
            }
            darr = newArr;
        }
        darr[count++] = value;
    }
}

public class Main {
    public static void main(String args[])
    {
        dynamicArray numbers = new dynamicArray(5);
        numbers.insert(2);
        numbers.insert(3);
        numbers.insert(4);
        numbers.insert(5);
        numbers.insert(6);
        numbers.insert(8);
        numbers.insert(9);
```

```
        numbers.displayshape();
    }
}
```

QUES4: PROGRAM AND OUTPUT

Main.java

```
class dynamicArray {
    int darr[],count;
    public dynamicArray(int length)
    {    darr = new int[length]; }
     public void displayshape()
    {
         int k = count/2;
            for( int i=0 ; i<=k ;i++)
            {
                for(int j = 0; j<count; j++ )
                {
                    if(darr[j]%darr[i] == 0  )
                        System.out.print(darr[j]+" ");
                }
                System.out.println();
            }
    }
    public void insert(int value)
    {
        if (darr.length == count) {
            int newArr[] = new int[2 * count];
            for (int i = 0; i < count; i++) {
                newArr[i] = darr[i];
            }
            darr = newArr;
        }
        darr[count++] = value;
    }
}

public class Main {
    public static void main(String args[])
    {
        dynamicArray numbers = new dynamicArray(5);
        numbers.insert(2);
        numbers.insert(3);
        numbers.insert(4);
        numbers.insert(5);
        numbers.insert(6);
        numbers.insert(8);
        numbers.insert(9);
        numbers.displayshape();
    }
```

QUES4: PROGRAM OUTPUT

```java
class dynamicArray {
    int darr[],count;
    public dynamicArray(int length)
    {    darr = new int[length]; }
     public void displayshape()
    {
          int k = count/2;
             for( int i=0 ; i<=k ;i++)
             {
                 for(int j = 0; j<count; j++ )
                 {
                     if(darr[j]%darr[i] == 0  )
                         System.out.print(darr[j]+" ");
                 }
                 System.out.println();
             }
    }
    public void insert(int value)
    {
        if (darr.length == count) {
            int newArr[] = new int[2 * count];
            for (int i = 0; i < count; i++) {
                newArr[i] = darr[i];
            }
            darr = newArr;
```

```
2 4 6 8
3 6 9
4 8
5


...Program finished with exit code 0
Press ENTER to exit console.
```

Ques 5: Write a java program to print following output (reverse should not be used)

Input : Lets learn java together

Output : Letslearnjavatogether

SOL:

```
import java.util.*;
public class Main
{
        public static void main(String args[])
        {
           Scanner sc = new Scanner(System.in);
           String str = sc.nextLine();
           char a[] = str.toCharArray();
    StringBuffer s = new StringBuffer();
    for (int i = 0; i < a.length; i++) {
       if ((a[i] != ' ') && (a[i] != '\t'))
          s.append(a[i]);
    }
    System.out.println(s.toString());
  }
}
```

QUES5: PROGRAM AND OUTPUT

Main.java

```java
import java.util.*;
public class Main
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        char a[] = str.toCharArray();
        StringBuffer s = new StringBuffer();
        for (int i = 0; i < a.length; i++) {
            if ((a[i] != ' ') && (a[i] != '\t'))
                s.append(a[i]);
        }
        System.out.println(s.toString());
    }
}
```

input

```
Lets learn java together
Letslearnjavatogether


...Program finished with exit code 0
Press ENTER to exit console.
```

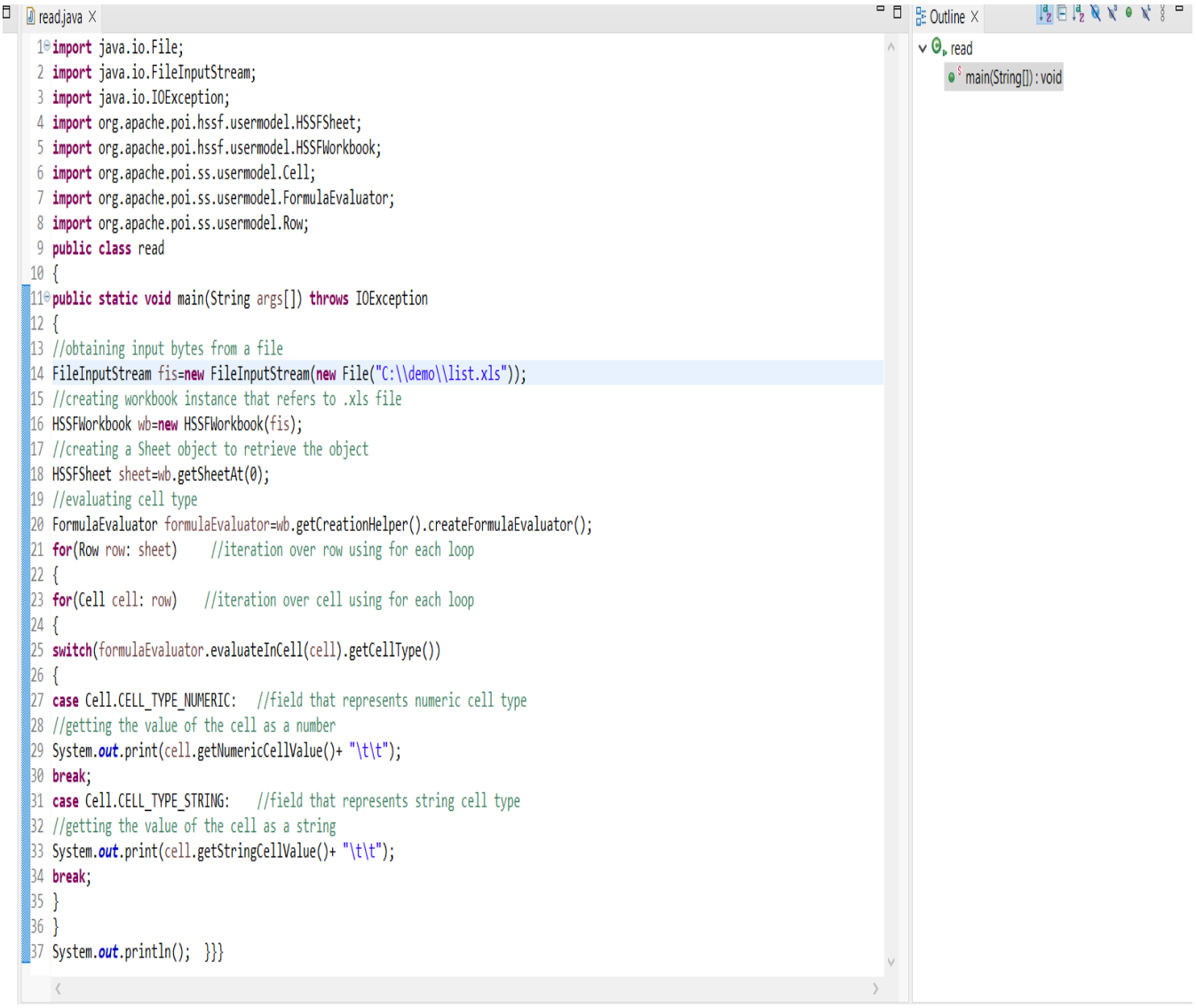
Ques 6: Write a java program to traverse data in a excel.

SOL: Read.java

```
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.FormulaEvaluator;
import org.apache.poi.ss.usermodel.Row;
public class Read
{
public static void main(String args[]) throws IOException
{
//obtaining input bytes from a file
FileInputStream fis=new FileInputStream(new File("C:\\demo\\list.xls"));
//creating workbook instance that refers to .xls file
HSSFWorkbook wb=new HSSFWorkbook(fis);
//creating a Sheet object to retrieve the object
HSSFSheet sheet=wb.getSheetAt(0);
//evaluating cell type
FormulaEvaluator formulaEvaluator=wb.getCreationHelper().createFormulaEvaluator();
for(Row row: sheet)     //iteration over row using for each loop
{
for(Cell cell: row)    //iteration over cell using for each loop
{
switch(formulaEvaluator.evaluateInCell(cell).getCellType())
{
case Cell.CELL_TYPE_NUMERIC:   //field that represents numeric cell type
//getting the value of the cell as a number
System.out.print(cell.getNumericCellValue()+ "\t\t");
```

```
break;
case Cell.CELL_TYPE_STRING:    //field that represents string cell type
//getting the value of the cell as a string
System.out.print(cell.getStringCellValue()+ "\t\t");
break;
}
}
System.out.println();
}
}
}
```

QUES 6: PROGRAM




```
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.FormulaEvaluator;
import org.apache.poi.ss.usermodel.Row;
public class read
{
public static void main(String args[]) throws IOException
{
//obtaining input bytes from a file
FileInputStream fis=new FileInputStream(new File("C:\\demo\\list.xls"));
//creating workbook instance that refers to .xls file
HSSFWorkbook wb=new HSSFWorkbook(fis);
//creating a Sheet object to retrieve the object
HSSFSheet sheet=wb.getSheetAt(0);
//evaluating cell type
FormulaEvaluator formulaEvaluator=wb.getCreationHelper().createFormulaEvaluator();
for(Row row: sheet)     //iteration over row using for each loop
{
for(Cell cell: row)    //iteration over cell using for each loop
{
switch(formulaEvaluator.evaluateInCell(cell).getCellType())
{
case Cell.CELL_TYPE_NUMERIC:   //field that represents numeric cell type
//getting the value of the cell as a number
System.out.print(cell.getNumericCellValue()+ "\t\t");
break;
case Cell.CELL_TYPE_STRING:    //field that represents string cell type
//getting the value of the cell as a string
System.out.print(cell.getStringCellValue()+ "\t\t");
break;
}
}
System.out.println();  }}}
```