JAVA PROGRAMMING
ASSIGNMENT 8

Name:PUNEETH L
USN: 1BM24MC069

1. Write a program that accepts a name list of five students from the command line and store in a vector.

Program:

```
import java.util.Vector;
public class StudentNames {
   public static void main(String[] args) {
      Vector<String> studentNames = new Vector<>();
      if (args.length != 5) {
         System.out.println("Please enter exactly 5 student names.");
         return;
      }
      for (String str:args) {
         studentNames.add(str);
      }
      System.out.println("Student Names:");
      for (String name : studentNames) {
         System.out.println(name);
      }
   }
}
```

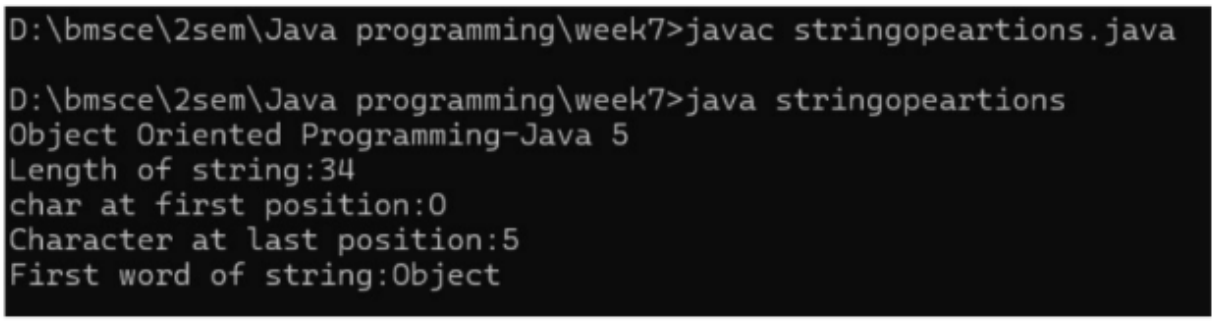
Output:

```
D:\bmsce\2sem\Java programming\8week>java StudentNames puneeth kumar aakash harish chethan
Student Names:
puneeth
kumar
aakash
harish
chethan
```

2. Write and run the following Java program that does the following:
a) Declare a string object named s1 containing the string "Object Oriented Programming- Java 5".
b) Print the entire string.
c) Use the length() method to find the length of the string.
d) Use the charAt() method to find the first character in the string.
e) Use charAt() and length() methods to print the last character in the string.
f) Use the indexOf() and the substring() method to print the first word in the String.
PROGRAM:

```
public class stringopeartions{
public static void main(String[] args) {
String s1="Object Oriented Programming-Java 5";
System.out.println(s1);
System.out.println("Length of string:"+s1.length());
System.out.println("char at first position:"+s1.charAt(0));
System.out.println("Character at last position:"+s1.charAt(s1.length()-1));
System.out.println("First word of string:"+s1.substring(0,s1.indexOf(" ")));
}
}
```

OUTPUT:



3. Define an exception called " No Equal Exception " that is thrown when a float value is not equal to 3.14. Write a program that uses the above user defined exception.

PROGRAM:

```
import java.util.*;
class NoEqualException extends Exception {
    NoEqualException(String message) {
      super(message);
   }
}

 class TestEqual {
   public static void main(String[] args) {
      Scanner sc= new Scanner(System.in);
      System.out.println("Enter a float number:");
      float value=sc.nextFloat();

      try {
         if (value != 3.14f) {
            throw new NoEqualException("Value is not equal to 3.14");
         } else {
            System.out.println("Value is equal to 3.14");
         }
      } catch (NoEqualException e) {
         System.out.println("Caught Exception: " + e.getMessage());
      }
   }
}
```

OUTPUT:

```
PS D:\bmsce\2sem\Java programming\week6> java TestEqual
Enter a float number:
3.14
Value is equal to 3.14
PS D:\bmsce\2sem\Java programming\week6> java TestEqual
Enter a float number:
2
Caught Exception: Value is not equal to 3.14
PS D:\bmsce\2sem\Java programming\week6>
```

4. Write a java program using threads to simulate traffic lights switch between Red, Green, and Yellow with fixed delays.

PROGRAM:

```
class TrafficLight extends Thread {
   String color;
   int delay;
   TrafficLight(String color, int delay) {
      this.color = color;
      this.delay = delay;
   }
   @Override
   public void run() {
      try {
         System.out.println(color + " light is ON");
         Thread.sleep(delay);
       } catch (InterruptedException e) {
         System.out.println("Interrupted: " + e.getMessage());
       }
   }
}
public class TrafficSimulation {
   public static void main(String[] args) {

         new TrafficLight("Red", 3000).start();
         try { Thread.sleep(3000); } catch (Exception e) {}
         new TrafficLight("Green", 2000).start();
         try { Thread.sleep(2000); } catch (Exception e) {}
         new TrafficLight("Yellow", 1000).start();
         try { Thread.sleep(1000); } catch (Exception e) {}
   }
}
```

OUTPUT:

```
PS D:\bmsce\2sem\Java programming\week6> java TrafficSimulation
Red light is ON
Green light is ON
Yellow light is ON
PS D:\bmsce\2sem\Java programming\week6>
```

5. Write a Java program to accept two parameters on the command line. If there are no command line arguments entered, the program should print the error message and exit. The program should check whether the first fi le exists and if it is an ordinary file. If it is so, then the content is copied to the second file.

Program:

```
import java.io.*;

public class Filecopy {
   public static void main(String[] args) {
      if (args.length != 2) {
         System.out.println("Error: Please provide two file names.");
         return;
      }
      File src = new File(args[0]);
      if (!src.exists() || !src.isFile()) {
         System.out.println("Error: Source file doesn't exist or is not a regular
file.");
         return;
      }

      try {
         FileInputStream in = new FileInputStream(src);
         FileOutputStream out = new FileOutputStream(args[1]);
         int ch;
         while ((ch = in.read()) != -1){
            out.write(ch);}
         in.close();
         out.close();
         System.out.println("File copied successfully.");
      } catch (IOException e) {
         System.out.println("Error: " + e);
      }
   }
}
```

Output:

```
D:\bmsce\2sem\Java programming\8week>cat inputfile.txt
this is a simple text file created for testing purpose of program written in java.
D:\bmsce\2sem\Java programming\8week>
D:\bmsce\2sem\Java programming\8week>cat outputfile
cat: outputfile: No such file or directory

D:\bmsce\2sem\Java programming\8week>java Filecopy inputfile.txt outputfile
File copied successfully.

D:\bmsce\2sem\Java programming\8week>cat outputfile
this is a simple text file created for testing purpose of program written in java.
D:\bmsce\2sem\Java programming\8week>
```