Siliguri Institute of Technology

Sukna, Siliguri-734009

Lab Manual

Object Oriented Programming with JAVA

Subject Code: **MCAN-293**

Bidyut Das
Assistant Professor
Department of MCA

Code: MCAN-293	Paper: Object Oriented Programming Lab using JAVA	
Contacts Hours / Week: 4	Total Contact Hours: 40	Credit: 2

Course Outcome:

After successful completion of this course, students will be able to:

- ✓ Apply object-oriented principles or features in software design process to develop Java programs for real life applications.
- ✓ Reduce the complexity of procedural language by employing different OOP technologies for developing robust and reusable software.
- ✓ Develop programs using stream classes for various I/O operations and design concurrent.
- ✓ Design graphical user interface to develop user interactive applications.

UNITS	COURSE CONTENT
1	Assignments on class, constructor, overloading, inheritance, overriding.
2	Assignments on wrapper class, arrays.
3	Assignments on developing interfaces- multiple inheritance, extending interfaces.
4	Assignments on creating and accessing packages.
5	Assignments on multithreaded programming
6	Assignments on applet programming

UNIT 1:

Setting the PATH variable to the appropriate directory location.

1. Setting the PATH variable temporarily:
   To set the temporary path of JDK, you need to follow the following steps:
   - Open the command prompt
   - Copy the path of the JDK/bin directory
   - Write in command prompt: set path="Path of the bin folder in java"
   - Example:

C:\>set path=C:\Program Files\Java\jdk1.6.0_23\bin

For setting the permanent path of JDK, you need to follow these steps:

- Go to MyPC properties -> advanced tab -> environment variables -> new tab of user variable -> write path in variable name -> write path of bin folder in variable value -> ok -> ok -> ok

Note: Take at least two classes to write a program. 1st class should hold all characteristics and behaviors and 2nd one will hold the main() .

e.g.

```
class A
{
        //All data and methods
}

class DemoA
{
public  static void main(String []args)
{
        //Create object reference/objects here
```

```
        //Call methods
    }
    }
```

- Save the program as DemoA.java
- Compile the program using javac DemoA.java
- Run the program using java DemoA

Sample Programs

PROGRAM 1

```
/* Write a Java Program to define a class, describe its constructor, overload the Constructors and instantiate its object */
import java.lang.*;

class student
{
String name; int regno;
int marks1,marks2,marks3;
// null constructor student()
{
name="raju"; regno=12345; marks1=56; marks2=47; marks3=78;
}
// parameterized constructor
student(String n,int r,int m1,int m2,int m3)
{
name=n; regno=r; marks1=m1; marks2=m2; marks3=m3;
}
// copy constructor student(student s)
{
name=s.name; regno=s.regno; marks1=s.marks1; marks2=s.marks2; marks3=s.marks3;
}
void display()
{
System.out.println(name + "\t" +regno+ "\t" +marks1+ "\t" +marks2+ "\t" + marks3);
}
}

class studentdemo
{
public static void main(String arg[])
{
student s1=new student();


student s2=new student("john",34266,58,96,84); student s3=new student(s1); s1.display(); s2.display();
s3.display();
}
}
```

**********OUTPUT**********

```
c:\jdk1.6.0_26\bin>javac studentdemo.java
```

```
c:\jdk1.6.0_26\bin>java studentdemo

raju    12345  56      47      78
john    34266  58      96      84
raju    12345  56      47      78
```

PROGRAM 2

/* Write a Java Program to define a class, define instance methods and overload them and use them for dynamic method invocation.*/

```
import java.lang.*;

class add
{
void display(int a,int b)
{
int c=a+b;
System.out.println("The sum of " + a + " & " + b + " is " + c);
}

void display(double a,double b)
{
double c=a+b;
System.out.println("The sum of " + a + " & " + b + " is " + c);
}
}

class add_demo
{
public static void main(String arg[])
{
add obj=new add(); obj.display(10,20); obj.display(10.2,20.2);
}
}
```

**********OUTPUT**********

```
c:\jdk1.6.0_26\bin>javac add_demo.java
c:\jdk1.6.0_26\bin>java add_demo

The sum of 10 & 20 is 30
The sum of 10.2 & 20.2 is 30.4
```

PROGRAM 3

/* Write a Java Program to demonstrate use of nested class.*/

```
import java.lang.*; class outer
{
int m=10; class inner
{
int n=20; void display()
{
System.out.println("m = "+m); System.out.println("n = "+n);
}
}
```

```
}

class nesteddemo
{
public static void main(String arg[])
{
outer outobj=new outer();
outer.inner inobj=outobj.new inner();
inobj.display();
}
}
```

**********OUTPUT**********

```
C:\jdk1.6.0_26\bin>javac nesteddemo.java C:\jdk1.6.0_26\bin>java nesteddemo
m = 10
n = 20
```

PROGRAM 4

Write a JAVA Program to implement Inner class and demonstrate its Access Protections.

```
class OuterClass
{
int a[]=new int[10];
int sum=0,i,k=0;
void initialize()
{
System.out.println("Inside Outer class method");
for(i=0;i<10;i++)
{
a[i]=k;
k++;
}
System.out.println("Array Elements are");
for(i=0;i<10;i++)
System.out.print(" " +a[i]);
System.out.println();
}
void show()
{
InnerClass ob=new InnerClass();
ob.cal();
}
class InnerClass
{
void cal()
{
System.out.println("Inside Inner class method");
for(i=0;i<10;i++)
sum+=a[i];
System.out.println("Sum :" +sum);
}
}
```

```
}
class Prog1b
{
public static void main(String args[])
{
OuterClass ob1=new OuterClass();
ob1.initialize();
ob1.show();
}
}
```

OUTPUT:
Inside Outer class method
Array elements are
0 1 2 3 4 5 6 7 8 9
Inside Inner class method
Sum : 45

Assignment on class, constructor, overloading, inheritance, overriding.

Assignment 1

1. Write a Java program that calculates factorial of a number (inputted via keyboard) recursively.
2. Write a program to swap two values using object reference. Your program should have a swap method.
3. Write java program to print Biggest of 3 Numbers using Logical Operators
4. Write a java program to print first 10 numbers in Fibonacci series
5. Write a java program to print Factorial of a given number
6. Write a java Program for swapping two numbers
7. Write a java program to print primes up to the given prime number
8. Write a java program to print sum of n terms in the series 1/1! +1/2!+1/3!.....
9. Write A Java Program to print Quadratic roots using command line arguments
10. Write a Java program that prints the season name corresponding to its month number using If-else and switch- case statements.\

Assignment 2

1. Write a program to create a class Student2 along with two method getData(),printData() to get the value through argument and display the data in printData. Create the two objects s1 ,s2 to declare and access the values from class STtest.

2. WAP using parameterized constructor with two parameters id and name. While creating the objects obj1 and obj2 passed two arguments so that this constructor gets invoked after creation of obj1 and obj2.

3. Write a program in java to generate an abstract class A also class B inherits the class A. generate the object for class B and display the text "call me from B".

4. Write a java program in which you will declare two interface sum and Add inherits these interface through class A1 and display their content.

5. Write a java program in which you will declare an abstract class Vehicle inherits this class from two classes car and truck using the method engine in both display "car has good engine" and "truck has bad engine".

6. Write a Java program that creates a Class, namely Student.

i.Ensure that Age instance variable of the Class is never accessed directly, and its value is never less than 4 and greater than 40 for any Object of the Class (use methods to validate and assign the value).

ii.Ensure that the constructor always assigns a unique value to Enrollment_No instance variable for every Object of the Class (use a static class variable for counting objects, say Object_Counter).

iii.Ensure that when an Object is removed, the Object_Counter is automatically decremented (use finalize()), and whenever required the variable can only be accessed using a method even without an Object reference (make the counter private and use a static method to access it).

7. Write a Java program that creates a Class namely A that has a private instance variable and method, a protected instance variable and method, a default instance variable and method, and a public instance variable and method. Create another Class say B that inherits from A.

i.Show that all except private members are inherited.

ii.Show that an inherited instance variable can be shadowed (with the same or weaker access visibility) but can be accessed using super keyword in the sub-class.

iii.Show that the reference variable of type A or B can't access an overridden method of A in the Object of B.

8. Write a JAVA Program to demonstrate Constructor overloading and Method overloading.
9. Write a JAVA Program to demonstrate Dynamic method dispatch.

UNIT 2:
[Wrapper class, arrays.]

```
//Java Program to print the array elements using for-each loop
class Testarray1{
public static void main(String args[]){
int arr[]={33,3,4,5};
//printing array using for-each loop
for(int i:arr)
System.out.println(i);
}}
```

Sample Programs

Program 1

```
//Java Program to demonstrate the way of passing an array
    //to method.
    class Testarray2{
    //creating a method which receives an array as a parameter
    static void min(int arr[]){
    int min=arr[0];
    for(int i=1;i<arr.length;i++)
     if(min>arr[i])
      min=arr[i];

    System.out.println(min);
    }

    public static void main(String args[]){
    int a[]={33,3,4,5};//declaring and initializing an array
    min(a);//passing array to method
    }}
```

Program 2

```
//Java Program to return an array from the method
class TestReturnArray{
//creating method which returns an array
static int[] get(){
return new int[]{10,30,50,90,60};
}

public static void main(String args[]){
//calling method which returns an array
int arr[]=get();
//printing the values of an array
for(int i=0;i<arr.length;i++)
System.out.println(arr[i]);
}}
```

Program 3

You are given a function,
int findCount(int arr[], int length, int num, int diff);

The function accepts an integer array 'arr', its length and two integer variables 'num' and 'diff'. Implement this function to find and return the number of elements of 'arr' having an absolute difference of less than or equal to 'diff' with 'num'.
Note: In case there is no element in 'arr' whose absolute difference with 'num' is less than or equal to 'diff', return -1.

Example:
Input:

arr: 12 3 14 56 77 13
num: 13
diff: 2

Output:
3

Explanation:
Elements of 'arr' having absolute difference of less than or equal to 'diff' i.e. 2 with 'num' i.e. 13 are 12, 13 and 14.

```
import java.util.*;
class Main
{
   public static int findCount (int arr[], int length, int num, int diff)
   {
      int count = 0;
      for (int i = 0; i < length; i++)
      {
         if (Math.abs (num - arr[i]) <= diff)
            count++;
      }
      return count>0?count:-1;
   }
```

```
    public static void main (String[]args)
    {
        Scanner sc = new Scanner (System.in);
        int n = sc.nextInt ();
        int arr[] = new int[n];

        for (int i = 0; i < n; i++)
            arr[i] = sc.nextInt ();
        int num = sc.nextInt ();
        int diff = sc.nextInt ();

        System.out.println (findCount (arr, n, num, diff));
    }
}
```

Program 4
Implement the following Function

def ProductSmallestPair(sum, arr)

The function accepts an integers sum and an integer array arr of size n. Implement the function to find the pair, (arr[j], arr[k]) where j!=k, Such that arr[j] and arr[k] are the least two elements of array (arr[j] + arr[k] <= sum) and return the product of element of this pair

NOTE

Return -1 if array is empty or if n<2
Return 0, if no such pairs found
All computed values lie within integer range

Example

Input

sum:9

Arr:5 2 4 3 9 7 1

Output 2

Explanation

Pair of least two element is (2, 1) 2 + 1 = 3 < 9, Product of (2, 1) 2*1 = 2. Thus, output is 2

Sample Input

sum:4

Arr:9 8 3 -7 3 9

Sample Output

```
import java.util.*;
class Main
{
   public static int productSmallestPair (int arr[], int n, int sum)
   {
      if (n <2)
         return -1;
      int ans, temp, check;
      for (int i = 0; i < n; i++)
      {
         for (int j = i + 1; j < n; j++)
            {
            if (arr[i] > arr[j])
               {
               temp = arr[i];
               arr[i] = arr[j];
               arr[j] = temp;
            }
         }
      }
      check = arr[0] + arr[1];

      if (check <= sum)
         return arr[0] * arr[1];
      else
         return 0;
   }

   public static void main (String[]args)
   {
      Scanner sc = new Scanner (System.in);
      int sum = sc.nextInt ();
      int n = sc.nextInt ();
      int arr[] = new int[n];

      for (int i = 0; i < n; i++)
         arr[i] = sc.nextInt ();
      System.out.println (productSmallestPair (arr, n, sum));
   }
}
```

Program 5

Write a Java Program to implement Wrapper classes and their methods. import java.io.*;

```
class wrapperdemo
{
public static void main(String args[])
{
Float P=new Float(0); Float I=new Float(0); int y=0;
try
{
DataInputStream ds=new DataInputStream(System.in); System.out.println("ENTER THE PRINCIPAL
```

```
AMOUNT"); System.out.flush();
String sp=ds.readLine(); P=Float.valueOf(sp);
System.out.println("ENTER THE INTEREST RATE"); System.out.flush();
String SI=ds.readLine(); I=Float.valueOf(SI);
System.out.println("ENTER THE NUMBER OF YEARS"); System.out.flush();
String sy=ds.readLine(); y=Integer.parseInt(sy);
}
catch(Exception e)
{
System.out.println("INPUT OUTPUT ERROR"); System.exit(1);
}
float value=loan(P.floatValue(),I.floatValue(),y); System.out.println("FINAL VALUE IS:"+value);
}
static float loan(float P,float I,int y)
{
int year=1; float sum=P; while(year<=y)
{
sum=sum+(P*I)/100; year++;
}
return sum;
}

}
```

**********OUTPUT**********

```
C:\jdk1.6.0_26\bin>javac wrapperdemo.java
Note: wrapperdemo.java uses or overrides a deprecated API. Note: Recompile with -Xlint:deprecation for details.

C:\jdk1.6.0_26\bin>java wrapperdemo ENTER THE PRINCIPAL AMOUNT 1000
ENTER THE INTEREST RATE
2
ENTER THE NUMBER OF YEARS 1
FINAL VALUE IS:1020.0

E:\jdk1.6.0_26\bin>java wrapperdemo ENTER THE PRINCIPAL AMOUNT 1000
ENTER THE INTEREST RATE 2
ENTER THE NUMBER OF YEARS 2
FINAL VALUE IS:1040.0
```

Assignment 3

1. Two Sum

Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

You can return the answer in any order.

Example 1:

Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].

Example 2:

Input: nums = [3,2,4], target = 6
Output: [1,2]

Example 3:

Input: nums = [3,3], target = 6
Output: [0,1]

---

2. 3Sum

Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0.

Notice that the solution set must not contain duplicate triplets.

Example 1:

Input: nums = [-1,0,1,2,-1,-4]
Output: [[-1,-1,2],[-1,0,1]]

Example 2:

Input: nums = []
Output: []

Example 3:

Input: nums = [0]
Output: []

3.

3Sum Closest

Given an integer array nums of length n and an integer target, find three integers in nums such that the sum is closest to target.

Return the sum of the three integers.

You may assume that each input would have exactly one solution.

Example 1:

Input: nums = [-1,2,1,-4], target = 1
Output: 2
Explanation: The sum that is closest to the target is 2. (-1 + 2 + 1 = 2).

Example 2:

Input: nums = [0,0,0], target = 1
Output: 0

---

4. Write a Java Program to implement array of objects.
5. Write a Java Program to implement bubble sort .
6. Write a Java Program to find the largest and smallest element from an array.
7. Write a Java Program to find addition of two matrices.

UNIT 3 & 4:

[Package & Interface]

Write a program to demonstrate use of implementing interfaces.

Program 1

```
import java.lang.*;

interface Area
{
final static float pi=3.14F; float compute(float x,float y);
}
class rectangle implements Area
{
public float compute(float x,float y)
{
return(pi*x*y);
}
}
class circle implements Area
{
public float compute(float x,float x)
{
return(pi*x*x);
}
}
class interfacedemo
{
public static void main(String a[])
{
rectangle rect=new rectangle(); circle cir=new circle();
Area A;
A=rect;
```

```
System.out.println("Area of rectangle="+A.compute(10,20)); A=cir;
System.out.println("Area of circle="+A.compute(30,0));
}
}
```

**********OUTPUT********** C:\jdk1.6.0_26\bin>javac interfacedemo.java
C:\jdk1.6.0_26\bin>java interfacedemo

Area of rectangle=628.0 Area of circle=2,827.43

PROGRAM 2

Write a program to demonstrate use of extending interfaces. import java.lang.*;

```
interface Area
{
final static float pi=3.14F;
double compute(double x,double y);
}

interface display extends Area
{
void display_result(double result);

}

class rectangle implements display
{
public double compute(double x,double y)
{
return(pi*x*y);
}
public void display_result(double result)
{
System.out.println("The Area is :" +result);
}
}

class InterfaceExtendsDemo
{
```

```
public static void main(String a[])
{
rectangle rect=new rectangle();
double result=rect.compute(10.2,12.3); rect.display_result(result);
}
}
```

**********OUTPUT**********

C:\jdk1.6.0_26\bin>javac InterfaceExtendsDemo.java C:\jdk1.6.0_26\bin>java InterfaceExtendsDemo
The Area is : 393.9444131612778

Program 3

Write a Java program to implement the concept of importing classes from user defined package and creating packages.

/*Source code of package p1 under the directory C:\jdk1.6.0_26\bin>p1\edit Student.java */

```
package p1;
public class Student
{
int regno; String name;
public void getdata(int r,String s)
{
regno=r; name=s;
}
public void putdata()
{
System.out.println("regno = " +regno); System.out.println("name = " + name);
}
}
```

/* Source code of the main function under C:\jdk1.6.0_26\bin>edit StudentTest.java */ import p1.*;

```
class StudentTest
{
public static void main(String arg[])
{
student s=new student(); s.getdata(123,"xyz"); s.putdata();
}
}
```

**********OUTPUT********** C:\jdk1.6.0_26\bin>javac p1\Student.java
C:\jdk1.6.0_26\bin>javac StudentTest.java

C:\jdk1.6.0_26\bin>java StudentTest

regno = 123

name = xyz

Try Yourself

1. Write a program to implement multiple inheritance in java.

String Handling

Sample Programs

Program 1

```
import java.lang.String;
class stringdemo
{
public static void main(String arg[])
{
String s1=new String("sit siliguri"); String s2="SIT SILIGURI";
System.out.println(" The string s1 is : " +s1); System.out.println(" The string s1 is : " +s2); System.out.println(" Length of the string s1 is : " +s1.length());
System.out.println(" The first accurence of r is at the position : " +s1.indexOf('r')); System.out.println(" The String in Upper Case : " +s1.toUpperCase()); System.out.println(" The String in Lower Case : " +s1.toLowerCase()); System.out.println(" s1 equals to s2 : " +s1.equals(s2));
System.out.println(" s1 equals ignore case to s2 : " +s1.equalsIgnoreCase(s2)); int result=s1.compareTo(s2);
System.out.println("After compareTo()"); if(result==0)
System.out.println( s1 + " is equal to "+s2); else if(result>0)
System.out.println( s1 + " is greather than to "+s2); else
System.out.println( s1 + " is smaller than to "+s2); System.out.println(" Character at an index of 6 is :" +s1.charAt(6)); String s3=s1.substring(4,12);
System.out.println(" Extracted substring is :"+s3);
System.out.println(" After Replacing g with a in s1 : " + s1.replace('g','a')); String s4=" This is a book ";
System.out.println(" The string s4 is :"+s4); System.out.println(" After trim() :"+s4.trim());
}
}
```

Output:

The string s1 is : sit siliguriThe string s1 is : SIT SILIGURI

Length of the string s1 is : 12

The first accurence of r is at the position : 10

The String in Upper Case : SIT SILIGURI

The String in Lower Case : sit siliguri

s1 equals to s2 : false

s1 equals ignore case to s2 : true

After compareTo()

```
sit siliguri is greather than to SIT SILIGURI
Character at an index of 6 is :l
Extracted substring is :siliguri
After Replacing g with a in s1 : sit siliauri
The string s4 is : This is a book
After trim() :This is a book
java -cp /tmp/OXTxChSrxN stringdemo
The string s1 is : sit siliguri
The string s1 is : SIT SILIGURI
Length of the string s1 is : 12
The first accurence of r is at the position : 10
 The String in Upper Case : SIT SILIGURI
 The String in Lower Case : sit siliguri
 s1 equals to s2 : false
 s1 equals ignore case to s2 : true
After compareTo()
sit siliguri is greather than to SIT SILIGURI
Character at an index of 6 is :lExtracted substring is :siliguri
After Replacing g with a in s1 : sit siliauri
The string s4 is : This is a book
After trim() :This is a book
```

Java program to find all permutations of a given String using recursion.

```
 * For example, given a String "XYZ", this program will print
 * all 6 possible permutations of
 * input e.g. XYZ, XZY, YXZ, YZX, ZXY, XYX

public class StringPermutations {

   public static void main(String args[]) {
      permutation("123");
   }


 /*
  * A method exposed to client to calculate permutation of String in Java.
  */
  public static void permutation(String input){
        permutation("", input);
  }

   /*
    * Recursive method which actually prints all permutations
    * of given String, but since we are passing an empty String
    * as current permutation to start with,
    * I have made this method private and didn't exposed it to client.
```

```
   */
  private static void permutation(String perm, String word) {
      if (word.isEmpty()) {
          System.err.println(perm + word);

      } else {
          for (int i = 0; i &lt; word.length(); i++) {
              permutation(perm + word.charAt(i), word.substring(0, i)
                      + word.substring(i + 1, word.length()));
          }
      }

  }
}
```

Output:
123
132
213
231
312
321

Program 2

Java program to find all permutations of a given String using recursion.

```
 * For example, given a String "XYZ", this program will print
 * all 6 possible permutations of
 * input e.g. XYZ, XZY, YXZ, YZX, ZXY, XYX

public class StringPermutations {

   public static void main(String args[]) {
      permutation("123");
   }


 /*
  * A method exposed to client to calculate permutation of String in Java.
  */
  public static void permutation(String input){
        permutation("", input);
  }

  /*
   * Recursive method which actually prints all permutations
   * of given String, but since we are passing an empty String
   * as current permutation to start with,
   * I have made this method private and didn't exposed it to client.
   */
```

```
  private static void permutation(String perm, String word) {
      if (word.isEmpty()) {
          System.err.println(perm + word);

      } else {
          for (int i = 0; i &lt; word.length(); i++) {
              permutation(perm + word.charAt(i), word.substring(0, i)
                      + word.substring(i + 1, word.length()));
          }
      }

  }
}
```

Output:
123
132
213
231
312
321

Assignment 4

1. Letter Combinations of a Phone Number

Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent. Return the answer in any order.
A mapping of digit to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.

Example 1:
Input: digits = "23"
Output: ["ad","ae","af","bd","be","bf","cd","ce","cf"]
Example 2:
Input: digits = ""
Output: []
Example 3:
Input: digits = "2"
Output: ["a","b","c"]

2.

Longest Substring Without Repeating Characters

Given a string s, find the length of the longest substring without repeating characters.
Example 1:

Input: s = "abcabcbb"
Output: 3
Explanation: The answer is "abc", with the length of 3.

Example 2:

Input: s = "bbbbb"
Output: 1
Explanation: The answer is "b", with the length of 1.

Example 3:

Input: s = "pwwkew"
Output: 3
Explanation: The answer is "wke", with the length of 3.
Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

---

3. **Longest Common Prefix**

Write a function to find the longest common prefix string amongst an array of strings.

If there is no common prefix, return an empty string "".

Example 1:

Input: strs = ["flower","flow","flight"]
Output: "fl"

Example 2:

Input: strs = ["dog","racecar","car"]
Output: ""
Explanation: There is no common prefix among the input strings.

---

4. **Password Checker**

You are given a function.
int CheckPassword(char str[], int n);
The function accepts string str of size n as an argument. Implement the function which returns 1 if given string str is valid password else 0.
str is a valid password if it satisfies the below conditions.

– At least 4 characters
– At least one numeric digit
– At Least one Capital Letter
– Must not have space or slash (/)
– Starting character must not be a number

Assumption:
Input string will not be empty.

Example:

Input:
aA1_67
Output:
1

Sample Input:
a987 abC012
Output:
0

UNIT 5

[Multithreaded Programming]

PROGRAM 1

Write a program to implement the concept of threading by extending Thread Class

```
import java.lang.Thread;
class A extends Thread
{
public void run()
{
System.out.println("thread A is sterted:"); for(int i=1;i<=5;i++)
{
System.out.println("\t from thread A:i="+i);
}
System.out.println("exit from thread A:");
}
}

class B extends Thread
{
public void run()
{
System.out.println("thread B is sterted:"); for(int j=1;j<=5;j++)
{
System.out.println("\t from thread B:j="+j);
}
System.out.println("exit from thread B:");
}
}

class C extends Thread
{
public void run()
{
System.out.println("thread C is sterted:"); for(int k=1;k<=5;k++)
{
System.out.println("\t from thread C:k="+k);
}
System.out.println("exit from thread C:");
```

```
}
}
class Threadtest
{
public static void main(String arg[])
{
new A().start();
new B().start();


new C().start();

}
}
```

**********OUTPUT**********

```
    thread A is sterted:
    thread B is sterted:
     thread C is started:
            from thread A:i=1
            from thread B:j=1
             from thread C:k=1
            from thread A:i=2
             from thread B:j=2
            from thread C:k=2
             from thread A:i=3
             from thread B:j=3
             from thread C:k=3
             from thread A:i=4
            from thread B:j=4
             from thread C:k=4
             from thread A:i=5
            from thread B:j=5
            from thread C:k=5
    exit from thread A:
    exit from thread B:
    exit from thread C:
```

PROGRAM 2

Write a program to implement the concept of threading by implementing Runnable Interface

```
import java.lang.Runnable;
class X implements Runnable
{
public void run()
{
for(int i=1;i<10;i++)
{
System.out.println("\t Thread X:"+i);
}
```

```
System.out.println("End of Thread X");
}
}

class Runnabletest
{
public static void main(String arg[])
{
X R=new X();
Thread T=new Thread(R); T.start();
}
}
```

**********OUTPUT**********

```
Thread X:1 Thread X:2
Thread X:3 Thread X:4
Thread X:5 Thread X:6
Thread X:7 Thread X:8
Thread X:9
End of Thread X
```

Program 3
Write a JAVA program using Synchronized Threads, which demonstrates Producer Consumer concept.

```
public class Prog4 {
public static void main(String[] args) {
Temp c = new Temp();
Producer p1 = new Producer(c, 1);
Consumer c1 = new Consumer(c, 1);
p1.start();
c1.start();
}
}
class Temp {
private int contents;
private boolean available = false;
public synchronized int get() {
while (available == false) {
try {
wait();
} catch (InterruptedException e) { }
}
available = false;
notifyAll();
return contents;
}
public synchronized void put(int value) {
while (available == true) {
try {
wait();
} catch (InterruptedException e) { }
}
```

```
contents = value;
available = true;
notifyAll();
}
}
class Consumer extends Thread {
private Temp t;
private int number;
public Consumer(Temp tmp, int number) {
t = tmp;
this.number = number;
}
public void run() {
int value = 0;
for (int i = 0; i < 10; i++) {
value = t.get();
System.out.println("Consumer #" + this.number+ " got: " + value);
}
}
}

class Producer extends Thread {
private Temp t;
private int number;
public Producer(Temp tmp, int number) {
t = tmp;
this.number = number;
}
public void run() {
for (int i = 0; i < 10; i++) {
t.put(i);
System.out.println("Producer #" + this.number
+ " put: " + i);
try {
sleep((int)(Math.random() * 100));
} catch (InterruptedException e) { }
}
}
}
```

OUTPUT:

```
Producer #1 put : 0
Consumer #1 got : 0
Producer #1 put : 1
Consumer #1 got : 1
Producer #1 put : 2
Consumer #1 got : 2
Producer #1 put : 3
Consumer #1 got : 3
Producer #1 put : 4
Consumer #1 got : 4
Producer #1 put : 5
Consumer #1 got : 5
```

Producer #1 put :6
Consumer #1 got : 6
Producer #1 put : 7
Consumer #1 got : 7
Producer #1 put : 8
Consumer #1 got : 8
Producer #1 put : 9
Consumer #1 got : 9

Assignment 5

1. Write a java program in which thread sleep for 5 sec and change the name of thread.
2. Write a java program for multithread in which user thread and thread started from main method invoked at a time each thread sleep for 1 sec.
3. Write a java program for to solve producer consumer problem in which a producer produces a value and consumer consume the value before producer generate the next value.
4. Write a java program for to solve printer synchronization problem in which all the jobs must be completed in order.

UNIT 6

[Applet]

Sample Programs

Program 1

Write a program using Applet to display a message in the Applet.

```
import java.applet.*; import
java.awt.Graphics;

/* <applet code="Appletdemo.class" width=300 height=300> </applet> */

public class Appletdemo extends Applet
{
  public void paint(Graphics g)
  {
    String msg="HELLO!, Welcome to my applet ";
    g.drawString(msg,80,150);
  }
}
```

**********OUTPUT**********

```
C:\jdk1.6.0_26\bin>javac Appletdemo.java

C:\jdk1.6.0_26\bin>appletviewer Appletdemo.java
```

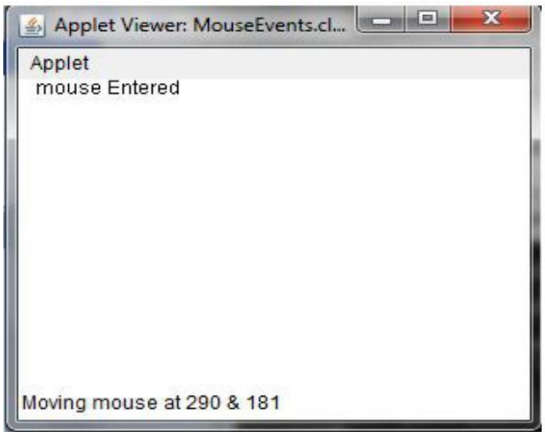
Program 2

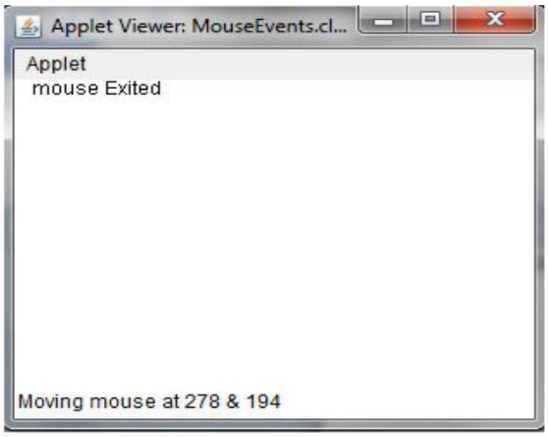
Write a Java Program to demonstrate Mouse events

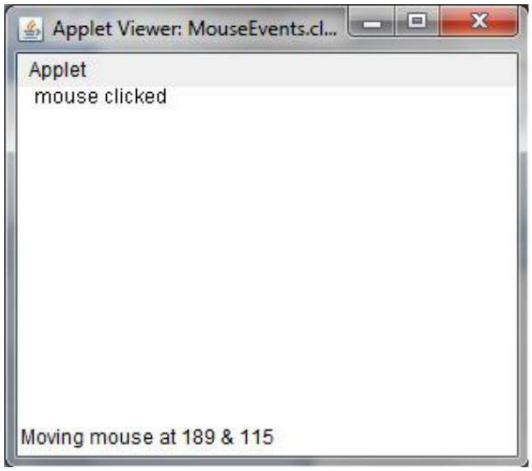
```
import java.applet.*; import
java.awt.event.*; import java.awt.*;

/* <applet code="MouseEvents.class" width=300 height=200> </applet> */

public class MouseEvents extends Applet implements
MouseListener,MouseMotionListener {
    String msg =" "; int
    x=0,y=0;
 public void init()
   {
     addMouseListener(this); addMouseMotionListener(this);
   }

public void mouseClicked(MouseEvent m)
   {
    x=10; y=10;
    msg ="mouse clicked"; repaint();
   }

public void mouseEntered(MouseEvent m)
   {
    x=10; y=10;
    msg ="mouse Entered"; repaint();
   }

public void mouseExited(MouseEvent m)
   {
    x=10; y=10;
    msg ="mouse Exited"; repaint();
   }
```

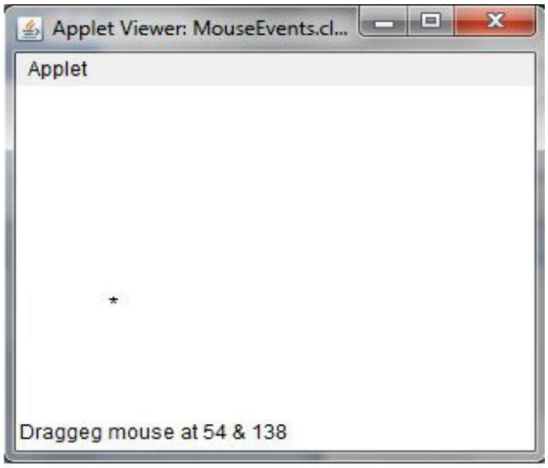
```
public void mousePressed(MouseEvent m)
   {
    x=m.getX();
    y=m.getY(); msg
    ="Down"; repaint();


   }
public void mouseReleased(MouseEvent m)
   {
    x=m.getX();
    y=m.getY();
    msg ="Up"; repaint();
    }

public void mouseDragged(MouseEvent m)
   {
    x=m.getX();
    y=m.getY();
    msg ="*";
    showStatus("Draggeg mouse at " +x+ " & "+y); repaint();
  }
public void mouseMoved(MouseEvent m)
   {
    showStatus("Moving mouse at " +m.getX()+ " & "+m.getY());
   }
public void paint(Graphics g)
   {
     g.drawString(msg,x,y);
    }
}
```

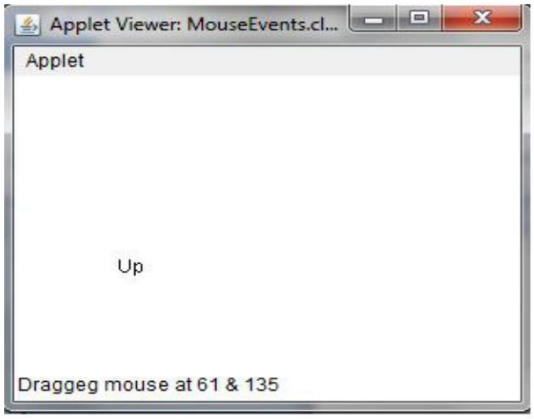
OUTPUT

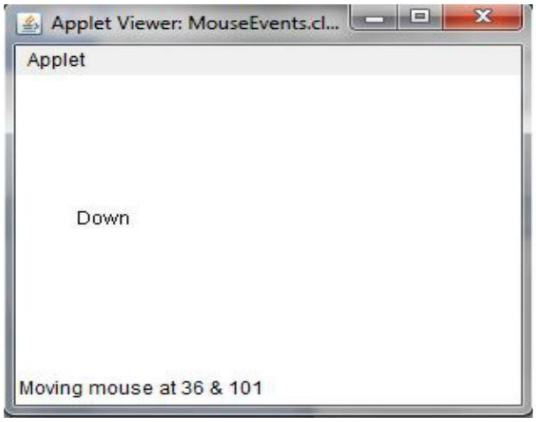






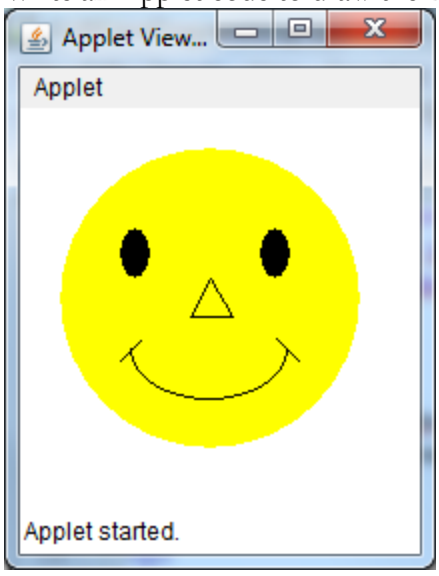






Assignment 6 [Applet]

1. Write an Applet code to draw the following smiley face



Exception Handling

PROGRAM 1
Write a program to implement the concept of Exception Handling using predefined exception.

```
import java.lang.*; class Exception_handle
{
public static void main(String argv[])
{
int a=10,b=5,c=5,x,y; try
{
x=a/(b-c);
}
catch(ArithmeticException e)
{
System.out.println("DIVISION BY ZERO");
}
y=a/(b+c); System.out.println("y="+y);
}
}
```

**********OUTPUT**********

```
DIVISION BY ZERO
y=1
```

PROGRAM 2

Write a program to implement the concept of Exception Handling by creating user defined exceptions.

```
import java.lang.Exception; import java.lang.*;
import java.lang.Exception;
import java.io.DataInputStream;

class MyException extends Exception
{
MyException(String message)
{
super(message);
}
}

class userdef
{
public static void main(String a[])
{
int age;
DataInputStream ds=new DataInputStream(System.in); try
{
System.out.println("Enter the age (above 15 abd below 25) :"); age=Integer.parseInt(ds.readLine()); if(age<15 ||
age> 25)

{
```

```
throw new MyException("Number not in range");
}
System.out.println(" the number is :" +age);
}

catch(MyException e)
{
System.out.println("Caught MyException"); System.out.println(e.getMessage());
}
catch(Exception e){ System.out.println(e); }

}
}
```

**********OUTPUT 1********** c:\jdk1.6.0_26\bin>java userdef
Enter the age (above 15 abd below 25) :
6
Caught MyException Number not in range

**********OUTPUT 2********** c:\jdk1.6.0_26\bin>java userdef
Enter the age (above 15 abd below 25) : 20
the number is :20

File Handling

Program 1
Write a JAVA Program which uses FileInputStream / FileOutPutStream Classes.

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
public class Prog8 {
public static void main(String[] args) throws Exception {
FileInputStream fis = new FileInputStream("in.txt");
FileOutputStream fos = new FileOutputStream("out.txt");
int c;
while ((c = fis.read()) != -1) {
System.out.print((char) c);
fos.write(c);
}
fis.close();
fos.close();
}
}
```

OUTPUT:
This is file input stream example program

Program 2
Write a JAVA Program which writes a object to a file (use transient variable also)

```
import java.io.*;
class Person implements Serializable {
private String firstName;
private String lastName;
private transient String password;
public Person() {
}
```

```
public String getFirstName() {
return firstName;
}
public void setFirstName(String firstName) {
this.firstName = firstName;
}
public String getLastName() {
return lastName;
}
public void setLastName(String lastName) {
this.lastName = lastName;
}
public String getPassword() {
return password;
}
public void setPassword(String pass) {
password=pass;
}
public String toString() {
StringBuffer buffer = new StringBuffer();
buffer.append(firstName);
buffer.append("\n");
buffer.append(lastName);
buffer.append("\n");
buffer.append(password);
buffer.append("\n");
return buffer.toString();
}
}

public class Prog9 {
public void writePersons(String filename) {
ObjectOutputStream outputStream = null;
ObjectInputStream ois=null;
try {
outputStream = new ObjectOutputStream(new FileOutputStream(filename));
Person person = new Person();
person.setFirstName("Guru");
person.setLastName("Prasad");
person.setPassword("aaa");
outputStream.writeObject(person);
outputStream.flush();
person = new Person();
person.setFirstName("Rama");
person.setLastName("Krishna");
person.setPassword("ccc");
outputStream.writeObject(person);
} catch (FileNotFoundException ex) {
ex.printStackTrace();
} catch (IOException ex) {
ex.printStackTrace();
} finally {
//Close the ObjectOutputStream
try {
if (outputStream != null) {
outputStream.flush();
outputStream.close();
}
} catch (IOException ex) {
ex.printStackTrace();
}
}
try {
Person p,q;
```

```
ois=new ObjectInputStream(new FileInputStream(filename));
p=(Person)ois.readObject();
System.out.println("Object read :" +p);
q=(Person)ois.readObject();
System.out.println("Object read :" +q);
ois.close();
}catch(Exception e)
{
e.printStackTrace();
}
}

public static void main(String[] args) {
new Prog9().writePersons("abc.txt");
}
}
```

OUTPUT:
Object read:
Guru
Prasad
null
Object read:
Rama
Krishna
null
/* Objects are written into the file serial.txt */

Assignment 7 [File]

1. Write a java program to create a file and write the text in it and save the file.
2. Write a java program to read a file and display the content on screen.
3. Write a java program to create a folder.
4. Write a java program to rename a file.
5. Write a java program in which data is read from one file and should be written in another file.name of both file is given through command line arguments.
6. Write a java program in which data is read from one file and should be written in another file line by line.
7. Write a java program to read the the file using BufferReader.

Bidyut Das
Assistant Professor
Department of Computer Application
Siliguri Institute of Technology