AMITY SCHOOL OF ENGINEERING & TECHNOLOGY

AMITY UNIVERSITY CAMPUS, SECTOR-125, NOIDA-201303

JAVA PROGRAMMING LAB

PRACTICAL FILE

COURSE CODE: IT 201

Submitted to:
Dr. Anshul Tickoo

Submitted by:
Shambhavi Mishra
4CSE-6Y

Index

S.No	Category of Assignment	Code	Name of Experiment	Date of Allotment of experiment	Date of Evaluation	Max. Marks	Marks obtained	Sign. of Faculty
1.	**Mandatory Experiment**	**LR (10)**	Write a program to print the given pattern 1 2 3 4 5 6			**1**		
2.			Distance travelled by a vehicle in t seconds is ut + ½*(a*t^2) Write a program to calculate the distance travelled by a vehicle at regular intervals of time given the values of acceleration and u. The program should provide the flexibility to the user to select their own time intervals and repeat calculations for different values of a and u.			**1**		
3.			An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in following figure. The figure also shows the minimum information required for each class. Specify all classes and define functions to create the database and retrieve individual information as and when required.			**1**		

4.			Write a program to create a class Student with data member roll number and functions getnumber and putnumber, class Test extends student with data members float m1,m2 and functions getmarks and putmarks, also create an interface Sports having static member float sports wt=6.0 and putwt, class result extends Test and implements Sports with data member total functions put wait and display.			**1**		
5.			An election is contested by 5 candidates the candidates are numbered 1-5 and voting is done by marking the candidate number on ballet paper. Write a program to read ballets and count the votes cast for each candidate using an array variable count. in case a number is outside the range 1-5 the ballet should be considered as spoilt ballet and program should also count the number of spoilt ballots.			**1**		
6.			Handle different types of exceptions a) Arithmetic Exception b) Array Index Out of Bounds c) Null Pointer d) Multiple catch block			**1**		

			e)Use of throws keyword. f)Use of finally block.					
7.			Write a program that reads in two floating point numbers and tests whether they are same up to three decimal places.			**1**		
8.			The daily maximum temperatures recorded in 2 cities during a week have been tabulated. Write a program to read the table elements into a two-dimensional array temperature, and to find the city and day corresponding to (a) the highest temperature and (b) the lowest temperature.			**1**		
9.			Multithreaded Programs a) Write a program to control the main thread. b) Write a program to create multiple threads.			**1**		
10.			Write a program to display the uses of priority in threads.			**1**		
11.			Write a program to show the use of yield (), stop () and sleep ().			**1**		
12.			Event Handling Programs. a) Write a program for handling Button Event b) Write a program for handling Mouse Event			**1**		
	OPEN ENDED EXPERIMENT	**PR (10)**				**10**		

EXPERIMENT 1

OBJECTIVE: Write a program to print the given pattern

```
1
2 3
4 5 6
```

SOFTWARE USED: NetBeans IDE

SOURCE CODE:

```
package javaapplication9;
import java.util.Scanner;
public class JavaApplication9 {
   public static void main(String[] args) {
      Scanner sc= new Scanner(System.in);
      int k=0;
      for(int i=1; i<=4;i++)
     {
        for (int j=1; j<=i;j++)
        {
           System.out.print(++k+" ");
        }
        System.out.println();
     }
   }

}
```

OUTPUT:

```
run:
1
2 3
4 5 6
7 8 9 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		

EXPERIMENT 2

OBJECTIVE: Distance travelled by a vehicle in t seconds is ut + ½*(a*t^2)
Write a program to calculate the distance travelled by a vehicle at regular intervals of time given the values of acceleration and u. The program should provide the flexibility to the user to select their own time intervals and repeat calculations for different values of a and u

SOFTWARE USED: NetBeans IDE

SOURCE CODE:

```
package javaapplication9;
import java.util.Scanner;
public class JavaApplication9 {
   public static void main(String[] args) {
     Scanner sc=new Scanner(System.in);
     double s;
     System.out.print("Enter the number of intervals=");
     int n= sc.nextInt();
     for(int i=1;i<=n;i++)
     {
        System.out.print("Enter the initial distance in metres=");
        float u= sc.nextFloat();
        System.out.print("Enter the time in seconds=");
        float t= sc.nextFloat();
        System.out.print("Enter the acceleration in m/s^2=");
        float a= sc.nextFloat();
        s=u*t+((1/2)*a*t*t);
         System.out.println("The total distance travelled=" +s);
     }
  }
  }
```

OUTPUT:

```
run:
Enter the number of intervals=2
Enter the initial distance in metres=2
Enter the time in seconds=10
Enter the acceleration in m/s^2=2
The total distance travelled=20.0
Enter the initial distance in metres=3
Enter the time in seconds=20
Enter the acceleration in m/s^2=4
The total distance travelled=60.0
```

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

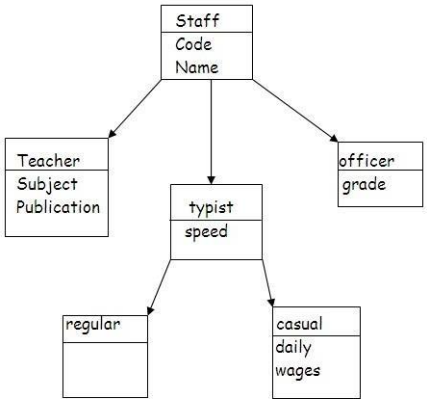
Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		

EXPERIMENT 3

OBJECTIVE: An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in following figure. The figure also shows the minimum information required for each class. Specify all classes and define functions to create the database and retrieve individual information as and when required.



SOFTWARE USED: NetBeans IDE

SOURCE CODE:

```
package javaapplication10;
import java.util.*;
class Staff
{
   int code;
   String name;
   Scanner sc= new Scanner(System.in);
   Staff ()
   {
      System.out.println ("Enter the code");
      code = sc.nextInt();
      System.out.println("Enter the name");
      name = sc.next();
      System.out.println ("The code is:" + code);
      System.out.println ("The Name is:" + name);
   }
}
class Teacher extends Staff
{
   int publication;
   Teacher ()
```

```
    {
      System.out.println ("Enter the publication");
        publication = sc.nextInt();
        System.out.println ("The publication is:" + publication);
    }

}
class Typist extends Staff{
    int speed;
    Typist (){
         System.out.println ("Enter the speed");
        speed = sc.nextInt();
        System.out.println ("The speed is:" + speed);

    }

}
class Officer extends Staff {
    char grade;
    Officer (){
         System.out.println ("Enter the grade");
        grade = sc.next().charAt(0);


          System.out.println ("The grade is:" + grade);
    }

}
class Casual extends Typist {
    int dailywages;
    Casual ()
    {
         System.out.println ("Enter the dailywages ");
        dailywages = sc.nextInt();
        System.out.println ("The dailywages is:" + dailywages);

    }
}
class Regular extends Typist{
int salary;
Regular()
```

```
{
   System.out.println ("Enter the salary ");
      salary = sc.nextInt();

      System.out.println ("The salary is:" + salary);
}
}
public class JavaApplication10 {
   public static void main(String[] args) {
       Scanner sc= new Scanner(System.in);
       int x;
       System.out.println ("Enter your choice");
       System.out.println ("1.Teacher \n 2.Typist \n 3.Officer");
       x = sc.nextInt();
       switch (x)
       {
          case 1:
             Teacher t=new Teacher();
             break;
          case 2:
             Typist ty=new Typist();
             break;
          case 3:
             Officer o=new Officer();
             break;
          default:
                break;
       }
     }
 }
```

OUTPUT:

```
Enter your choice
1.Teacher
 2.Typist
 3.Officer
1
Enter the code3
Enter the name Shambhavi
Shambhavi
The code is:3
The Name is:Shambhavi
Enter the publication
12
The publication is:12
```

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		

EXPERIMENT 4

OBJECTIVE: WAP to create a class Student with data member roll number and functions getnumber and putnumber, class Test extends student with data members float m1, m2 and functions getmarks and putmarks, also create an interface Sports having static member float sports wt=6.0 and putwt, class result extends Test and implements Sports with data member total functions put wait and display.

SOFTWARE USED: NetBeans IDE

SOURCE CODE:

```
package javaapplication8;
import java.util.*;

class student{
   int rollno;
   public void putno(){
      Scanner sc=new Scanner(System.in);
      System.out.print("Enter Roll No.:");
      rollno= sc.nextInt();
   }
   public void getno(){
      System.out.println(">>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<");
      System.out.println("The roll number is:" +rollno);
      System.out.println(">>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<");
   }
}
class test extends student{
   int m;
   int n;
   public void putmarks()
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter Marks of Mathematics:");
      m =sc.nextInt();
      System.out.println("Enter Marks of Science:");
      n =sc.nextInt();
   }
   public void getmarks(){
      System.out.println(">>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<");
```

```
        System.out.println("Marks of Mathematics are:" +m1);
        System.out.println("Marks of Science are:" +m2);
        System.out.println(">>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<");
    }
}
interface sports{
    static final int sportwt =6;
    public void putwt();
}
class result extends test implements sports
{

    public void putwt()
    {
        System.out.println("The Sport Weight is:"+sportwt);
    }
    public void display()
    {
        int total= m1+m2+sportwt;
        System.out.println(">>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<");
        System.out.println("Total Marks are="+total);
    }
}
public class JavaApplication8 {
    public static void main(String[] args) {
        result r=new result();
        r.putno();
        r.getno();
        r.putmarks();
        r.getmarks();
        r.putwt();
        r.display();
    }
}
```

OUTPUT:

```
Enter Roll No.:45
>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<
The roll number is:45
>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<
Enter Marks of Mathematics:
87
Enter Marks of Science:
90
>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<
Marks of Mathematics are:87
Marks of Science are:90
>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<
The Sport Weight is:6
>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<
Total Marks are=183
BUILD SUCCESSFUL (total time: 7 seconds)
```

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		

EXPERIMENT 5

OBJECTIVE: An election is contested by 5 candidates the candidates are numbered 1-5 and voting is done by marking the candidate number on ballet paper. Write a program to read ballets and count the votes cast for each candidate using an array variable count. in case a number is outside the range 1-5 the ballet should be considered as spoilt ballet and program should also count the number of spoilt ballots.

SOFTWARE USED: NetBeans IDE

SOURCE CODE:

```
package javaapplication8;
import java.util.Scanner;

public class JavaApplication8 {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n, ch, one = 0, two = 0, three = 0, four = 0, five = 0, x = 0;

    System.out.println("Enter total number of voters: ");
    n = sc.nextInt();

    for (int i = 0; i < n; i++) {
      System.out.println("\nEnter votes for:");
      System.out.println("1 2 3 4 5");
      System.out.println("\nYour vote goes to: ");
      ch = sc.nextInt();

      switch (ch) {
        case 1:
        one++;
        break;
        case 2:
        two++;
        break;
        case 3:
        three++;
        break;
        case 4:
```

```
        four++;
       break;
       case 5:
       five++;
       break;
       default:
       x++;
         break;
     }
   }

   System.out.println("\n Votes given to person 1 is: " + one);
   System.out.println("\n Votes given to person 2 is: " + two);
   System.out.println("\n Votes given to person 3 is: " + three);
   System.out.println("\n Votes given to person 4 is: " + four);
   System.out.println("\n Spoilt votes are: " + x);
   sc.close();
  }
}
```

OUTPUT:

```
Enter total number of voters: 3
Enter votes for:
1 2 3 4 5

Your vote goes to:
2
Enter votes for:
1 2 3 4 5

Your vote goes to:
3
Enter votes for:
1 2 3 4 5

Your vote goes to:
5
Votes given to person 1 is: 0
Votes given to person 2 is: 1
Votes given to person 3 is: 1
Votes given to person 4 is: 0
Spoilt votes are: 1
```

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		

EXPERIMENT 6

OBJECTIVE: Handle different types of exceptions
a) Arithmetic Exception
b) Array Index Out of Bounds
c) Null Pointer
d) Multiple catch block
e) Use of throws keyword.
f) Use of finally block.

SOFTWARE USED: NetBeans IDE

SOURCE CODE:

a)

```
package javaapplication20;
import java.util.Scanner;
import java.lang.ArithmeticException;
public class JavaApplication20 {
        public static void main(String[] args) {
           Scanner sc = new Scanner(System.in);
           System.out.print("Enter Dividend:");
           int b = sc.nextInt();
           System.out.print("Enter Divisor:");
           int a = sc.nextInt();
           int c ;
           try
           {
              c = b/a;
              System.out.println(">>>>>>>Try Block<<<<<<<<<");
              System.out.println("Quotient is" +c);
           }
           catch(ArithmeticException e)
                 {
                    System.out.println("Division by Zero");
                    System.out.println("Exception Caught in Catch Block");
                 }
   }
}
```

b)

```
package arrayoutofbounds;
import java.util.*;
public class Arrayoutofbounds {
    public static void main(String[] args)
    {
        int n;
        Scanner sc=new Scanner(System.in);
```

```
        int[] array = new int[5];
        try {
            System.out.print("Enter the number of elements you want to store: ");
            n=sc.nextInt();
            System.out.println("Enter the elements of the array: ");
            for(int i=0; i<n; i++)
            {
                array[i]=sc.nextInt();
            }
        }
        catch (Exception e) {
            System.out.println("\nException caught" +e);
        }
    }
    }

c)
package javaapplication14;
class M
{
    static void demoproc ()
    {
        try
        {
            throw new NullPointerException("demo");
        }
        catch(NullPointerException e)
        {
            System.out.println("Caught an Exception inside first catch block");
            throw e;
        }

    }
}
public class JavaApplication14 {
    public static void main(String[] args) {
        M m=new M();
        try
        {
             m.demoproc ();
        }
       catch (NullPointerException e)
       {
          System.out.println("Recaught:"+e);
       }
    }
}
```

d)

```
package javaapplication14;
class M
{
   static void demoproc ()
   {
      try
      {
         throw new NullPointerException("demo");
      }
      catch(NullPointerException e)
      {
         System.out.println("Caught an Exception inside first catch block");
         throw e;
      }
   }
}

public class JavaApplication14 {
   public static void main(String[] args) {
      M m=new M();
      try
      {
          m.demoproc ();
      }
     catch (NullPointerException e)
     {
        System.out.println("Recaught:"+e);
     }
   }
}
```

e)

```
class Test
{
        static void fun() throws IllegalAccessException
        {
                System.out.println("Inside the function");
                throw new IllegalAccessException("Illegal Exception");
        }
        public static void main(String args[])
        {
                try
                {
                        fun();
                }
                catch(IllegalAccessException e)
                {
```

```
                System.out.println(e);
            }
        }
}
```

f)

```
import java.io.*;
class M {
    public static void main(String[] args)
    {
        try {
            System.out.println("Inside try

            block");System.out.println(34 / 0);

    }
        catch (NullPointerException e) {
            System.out.println("catch : exception not handled.");
        }
        finally {
            System.out.println(
                "finally : i will execute always.");
        }
        System.out.println("i want to run");
    }
}
```

OUTPUTS:

a)

```
run:
Enter Dividend:12
Enter Divisor:0
Division by Zero
Exception Caught in Catch Block
BUILD SUCCESSFUL (total time: 8 seconds)
```

b)

```
Enter the number of elements you want to store: 6
Enter the elements of the array:
12 34 56 78 90 23

Exception caughtjava.lang.ArrayIndexOutOfBoundsException: 5
BUILD SUCCESSFUL (total time: 3 minutes 56 seconds)
```

c)

```
run:
Caught an Exception inside first catch block
Recaught:java.lang.NullPointerException: demo
BUILD SUCCESSFUL (total time: 0 seconds)
```

d)

```
run:
Divide by zerojava.lang.ArithmeticException: / by zero
BUILD SUCCESSFUL (total time: 0 seconds)
run:
Caught an Exception inside first catch block
Recaught:java.lang.NullPointerException: demo
BUILD SUCCESSFUL (total time: 0 seconds)
```

e)

```
Inside the function
java.lang.IllegalAccessException: Illegal Exception
```

f)

```
Inside try blockfinally : i will execute always.
Exception in thread "main" java.lang.ArithmeticException: / by zero
at M.main(M.java:9)
```

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		

EXPERIMENT 7

OBJECTIVE: Write a program that reads in two floating point numbers and tests whether they are same up to three decimal places.

SOFTWARE USED: NetBeans IDE

SOURCE CODE:

```
package javaapplication13;
import java.util.*;

public class JavaApplication13 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter floating number 1: ");
        float x = in.nextFloat();
        System.out.print("Enter floating number 2: ");
        float y = in.nextFloat();
        x = x * 1000;
        int p = (int)x;
        y = y * 1000;
        int q = (int)y;
        if (p == q)
        {
            System.out.println("Same up to three decimal places");
        }
        else
        {
            System.out.println("Different");
        }
    }
}
```

OUTPUT:

```
run:
Enter floating number 1: 23.45678
Enter floating number 2: 23.45689
Same up to three decimal places
BUILD SUCCESSFUL (total time: 11 seconds)
```

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		

EXPERIMENT 8

OBJECTIVE: The daily maximum temperatures recorded in 2 cities during a week have been tabulated. Write a program to read the table elements into a two-dimensional array temperature, and to find the city and day corresponding to
(a) the highest temperature and
(b) the lowest temperature.

SOFTWARE USED: NetBeans IDE

SOURCE CODE:

```
package javaapplication12;
import java.util.*;
public class JavaApplication12 {
  public static void main(String[] args) {
     int i,j;
     float temp[][]= new float[2][7];
     Scanner sc = new Scanner(System.in);
     System.out.println("Enter the temperature of 2 cities in one week");
     for(i=0;i<2;i++)
     {
        for(j=0;j<7;j++)
        {
           System.out.print("Enter temperature of City " +(i+1)+ " for Day " +(j+1)+": ");
           temp[i][j] = sc.nextFloat();
        }
     }
     int p=0,q=0,x=0,y=0;
     for(i=0;i<2;i++)
     {
        for(j=0;j<7;j++)
        {
           if (temp[i][j] > temp[p][q])
           {
              p=i;
              q=j;
           }
           if (temp[i][j]< temp[x][y])
           {
```

```
                x=i;
                y=j;
            }
        }
    }
    System.out.println("Highest Temperature:" +temp[p][q]);
    System.out.println("Lowest Temperature:" +temp[x][y]);
    System.out.println("City " +(p+1)+ " Day " +(q+1)+ " is
    the hottest day");
    System.out.println("City " +(x+1)+ " Day " +(y+1)+ " is
    the coldest day");
  }
}
```

OUTPUT:

```
Enter the temperature of 2 cities in one week
Enter temperature of City 1 for Day 1: 23.4
Enter temperature of City 1 for Day 2: 25.6
Enter temperature of City 1 for Day 3: 27.8
Enter temperature of City 1 for Day 4: 29.1
Enter temperature of City 1 for Day 5: 31.2
Enter temperature of City 1 for Day 6: 33.4
Enter temperature of City 1 for Day 7: 34.5
Enter temperature of City 2 for Day 1: 33.6
Enter temperature of City 2 for Day 2: 36.7
Enter temperature of City 2 for Day 3: 32.4
Enter temperature of City 2 for Day 4: 31.5
Enter temperature of City 2 for Day 5: 20.8
Enter temperature of City 2 for Day 6: 25.7
Enter temperature of City 2 for Day 7: 23.6
Highest Temperature:36.7
Lowest Temperature:20.8
City 2 Day 2 is the hottest day
City 2 Day 5 is the coldest day
```

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		

EXPERIMENT 9

OBJECTIVE: Multithreaded Programs
a) Write a program to control the main thread.
b) Write a program to create multiple threads.

SOFTWARE USED: NetBeans IDE

SOURCE CODE:
a)

```
package multithread;
class Multithread
{
   public static void main(String args[])
{
   Thread t = Thread.currentThread();
   System.out.println("Current thread: " + t);
   t.setName("My Thread");
   System.out.println("After name change: " + t);
   try
   {
      for(int n = 5; n > 0; n--)
      {
         System.out.println(n);
         Thread.sleep(10000);
      }
   }
   catch (InterruptedException e)
   {
      System.out.println("Main thread interrupted");
   }
}
} package multithread;
class Multithread
{
   public static void main(String args[])
{
   Thread t = Thread.currentThread();
   System.out.println("Current thread: " + t);
   t.setName("My Thread");
   System.out.println("After name change: " + t);
   try
   {
      for(int n = 5; n > 0; n--)
      {
         System.out.println(n);
```

```
            Thread.sleep(10000);
        }
    }
    catch (InterruptedException e)
    {
        System.out.println("Main thread interrupted");
    }
}
}
```

b)

```
package threaddemo;
class NewThread implements Runnable
{
  String name;
  Thread t;
NewThread(String threadname)
{
  name = threadname;
  t = new Thread(this, name);
  System.out.println("New thread: " + t);
  t.start();
}
public void run()
{
 try
 {
   for(int i = 5; i > 0; i--)
   {
   System.out.println(name + ": " + i);
   Thread.sleep(1000);
   }
 }
 catch (InterruptedException e)
 {
   System.out.println(name + "Interrupted");
 }
   System.out.println(name + " exiting.");
}
}
class ThreadDemo
{
public static void main(String args[])
{
  new NewThread("One");
  new NewThread("Two");
  new NewThread("Three");
 try
```

```
 {
  Thread.sleep(10000);
 }
 catch (InterruptedException e)
 {
  System.out.println("Main thread Interrupted");
 }
System.out.println("Main thread exiting.");
}
}
```

OUTPUTS:

a)

```
run:
Current thread: Thread[main,5,main]
After name change: Thread[My Thread,5,main]
5
4
3
2
1
BUILD SUCCESSFUL (total time: 50 seconds)
```

b)

```
run:
New thread: Thread[One,5,main]
New thread: Thread[Two,5,main]
New thread: Thread[Three,5,main]
One: 5
Two: 5
Three: 5
Two: 4
One: 4
Three: 4
Two: 3
One: 3
Three: 3
Two: 2
One: 2
Three: 2
Two: 1
One: 1
Three: 1
Two exiting.
One exiting.
Three exiting.
Main thread exiting.
BUILD SUCCESSFUL (total time: 10 seconds)
```

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		

EXPERIMENT 10

OBJECTIVE: Write a program to display the uses of priority in threads.

SOFTWARE USED: NetBeans IDE

SOURCE CODE:

```
package javaapplication22;
import java.lang.Exception;

class A extends Thread
{
   public void run()
   {
      System.out.println("Thread A started");
      for(int i=1;i<=4;i++)
      {
         System.out.println("\tFrom Thread a : i = " +i);
      }
      System.out.println("Exit from A");
   }
}

class B extends Thread
{
   public void run()
   {
      System.out.println("Thread B started");
      for(int j=1;j<=4;j++)
      {
         System.out.println("\tFrom Thread b : j = " +j);
      }
      System.out.println("Exit from B");
   }
}

class C extends Thread
{
   public void run()
   {
      System.out.println("Thread C started");
      for(int k=1;k<=4;k++)
```

```
        {
            System.out.println("\tFrom Thread c : k = " +k);

        }
        System.out.println("Exit from C");
    }
}

public class JavaApplication22 {
    public static void main(String args[ ])
    {
        A threadA = new A();
        B threadB = new B();
        C threadC = new C();

        threadC.setPriority(Thread.MAX_PRIORITY);
        threadB.setPriority(threadA.getPriority()+1);
        threadA.setPriority(Thread.MIN_PRIORITY);

        System.out.println("Start thread A");
        threadA.start();

        System.out.println("Start thread B");
        threadB.start();

        System.out.println("Start thread C");
        threadC.start();

        System.out.println("End of the main thread");
    }
}
```

OUTPUT:

```
Start thread A
Start thread B
Start thread C
End of the main thread
Thread C started
        From Thread c : k = 1
        From Thread c : k = 2
        From Thread c : k = 3
        From Thread c : k = 4
Exit from C
Thread B started
        From Thread b : j = 1
        From Thread b : j = 2
        From Thread b : j = 3
        From Thread b : j = 4
Exit from B
Thread A started
        From Thread a : i = 1
        From Thread a : i = 2
        From Thread a : i = 3
        From Thread a : i = 4
Exit from A
BUILD SUCCESSFUL (total time: 1 second)
```

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		

EXPERIMENT 11

OBJECTIVE: Write a program to show the use of yield (), stop () and sleep ().

SOFTWARE USED: NetBeans IDE

SOURCE CODE:

```
package javaapplication24;

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

```
}
public class JavaApplication24
{
  public static void main(String args[])
  {
    X runnable = new X();
    Thread threadX = new Thread(runnable);
    Y runnable1 = new Y();
    Thread threadY = new Thread(runnable1);
    Z runnable2 = new Z();
    Thread threadZ = new Thread(runnable2);
    threadX.start();
    threadY.start();
    threadZ.start();

    System.out.println("End of main thread");
  }
}
```

OUTPUT:

```
run:
End of main thread
        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
        Thread X : 10
End of Thread X
         Thread  Y:1
         Thread  Y:2
         Thread  Y:3
         Thread  Y:4
         Thread  Y:5
         Thread  Y:6
         Thread  Y:7
         Thread  Y:8
         Thread  Y:9
         Thread  Y:10
End of Thread Y
        Thread Z : 1
        Thread Z : 2
        Thread Z : 3
        Thread Z : 4
        Thread Z : 5
        Thread Z : 6
        Thread Z : 7
        Thread Z : 8
        Thread Z : 9
        Thread Z : 10
End of Thread Z
BUILD SUCCESSFUL (total time: 0 seconds)
```

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		

EXPERIMENT 12

OBJECTIVE: Event Handling Programs.
(a) Write a program for handling Button Event
(b) Write a program for handling Mouse Event

SOFTWARE USED: NetBeans IDE

SOURCE CODE:

a)

```
package buttondemo;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/* <applet code="ButtonDemo" width=250 height=150>
   </applet>
*/
public class ButtonDemo extends Applet implements ActionListener{
   String msg = "";
   Button yes,no,maybe;

   public void init() {
      yes = new Button("Yes");
      no =new Button("No");
      maybe = new Button("Undecided");

      add(yes);
      add(no);
      add(maybe);

      yes.addActionListener(this);
      no.addActionListener(this);
      maybe.addActionListener(this);
   }
   public void actionPerformed(ActionEvent ae){
      String str = ae.getActionCommand();
      if (str.equals("Yes")){
         msg="You pressed Yes.";
      }
      else if (str.equals("No")){
         msg="You pressed No.";
```

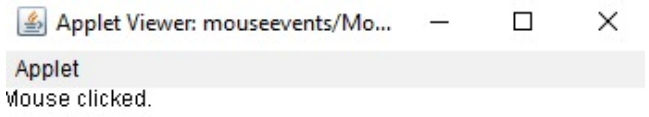
```
        }
        else {
           msg="You pressed Maybe.";
        }
        repaint();
     }
     public void paint(Graphics g){
        g.drawString(msg, 6, 100);
     }
}

b)
package mouseevents;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/* <applet code="MouseEvents" width=300 height=100>
   </applet>
*/
public class MouseEvents extends Applet implements MouseListener, MouseMotionListener {
   String msg = "";
   int mouseX=0, mouseY=0;
   public void init() {
      addMouseListener(this);
      addMouseMotionListener(this);
   }
   public void mouseClicked(MouseEvent me){
      mouseX = 0;
      mouseY = 10;
      msg = "Mouse clicked.";
      repaint();
   }
   public void mouseEntered(MouseEvent me){
      mouseX = 0;
      mouseY = 10;
      msg = "Mouse entered.";
      repaint();
   }
   public void mouseExited(MouseEvent me){
      mouseX = 0;
      mouseY = 10;
```

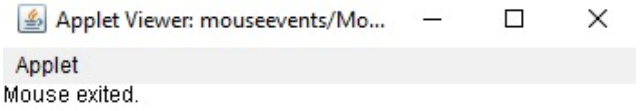
```
        msg = "Mouse exited.";
        repaint();
    }
    public void mousePressed(MouseEvent me){
        mouseX = me.getX();
        mouseY = me.getY();
        msg = "Down";
        repaint();
    }
    public void mouseReleased(MouseEvent me){
        mouseX = me.getX();
        mouseY = me.getY();
        msg = "Up";
        repaint();
    }
    public void mouseDragged(MouseEvent me){
        mouseX = me.getX();
        mouseY = me.getY();
        msg = "*";
        showStatus("Dragging mouse at "+ mouseX +","+ mouseY);
        repaint();
    }
    public void mouseMoved(MouseEvent me){
        showStatus("Moving mouse at "+ me.getX() +","+ me.getY());
    }
    public void paint(Graphics g){
        g.drawString(msg, mouseX, mouseY);
    }
}
```

OUTPUTS:

a)

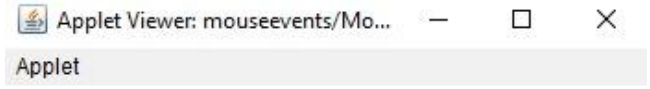
b)

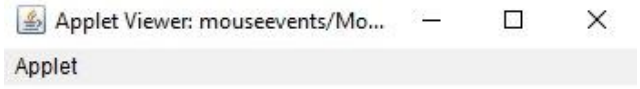




Moving mouse at 105,74

Moving mouse at 42,2



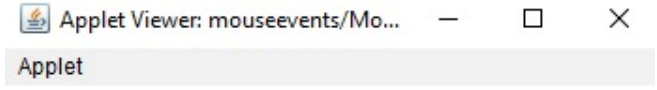


Down

Up

Moving mouse at 151,151

Dragging mouse at 96,77



*

Dragging mouse at 113,89

Internal Assessment (Mandatory Experiment) Sheet for Lab Experiment
Department of Computer Science & Engineering
Amity University, Noida (UP)

Program	B. Tech CSE	Course Name	Java Programming Lab
Course Code	IT201	Semester	IV
Student Name	Shambhavi Mishra	Enrollment Number	A2305221660

Marking Criteria

Criteria	Total Marks	Marks Obtained	Comments
Concept(A)	2		
Implementation(B)	2		
Performance(C)	2		
Total	6		