



JB INSTITUTE OF ENGINEERING & TECHNOLOGY

UGC Autonomous

Accredited by NAAC & NBA, Approved by AICTE & Permanently Affiliated to JNTUH



OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

**B.Tech AIML - R20
Regulation**



**Department of Artificial Intelligence
& Machine Learning**

Name of the Student	
Roll Number	
Class & Section	

Department of Artificial Intelligence &
Machine Learning

S.No	Name of the Experiment	Date of Completion	Faculty Signature
1	Write a Java program that implements the following a.Constructor b.Parameterized Constructor c.Method Overloading d.Constructor overloading		
2	a.Write a program that checks whether a given string is a palindrome or not Ex:MADAM is a palindrome b.Write a Java program for sorting a given list of names in ascending order. c.Write a Java program that reads a line of integers, and then displays each integer and the sum of all the integers(Use String Tokenizer class of java.util)		
3	Write a Java program that uses the following Keywords a.this b.superb c.static d.final		
4	a.Write a Java program to implement method overriding. b. Write a Java program to implement dynamic method dispatch. c. Write a Java program to implement multiple inheritance. d. Write a Java program that uses access specifiers.		
5	a. Write a Java program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes. b. Write a Java program that reads a file and displays the file on the screen, with a line number before each line. c. Write a Java program that displays the number of characters, lines and words in a text file		
6	a. Write a Java program for handling Checked Exceptions. b. Write a Java program for handling Unchecked Exceptions		
7	a. Write a Java program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds. b. Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.		
8	a. Develop an applet that displays a simple message. b. Develop an applet that receives an integer in one text field, and computes its factorial value and returns it in another text field, when the button named “Compute” is clicked		
9	Write a Java program that works as a simple calculator. Use a grid layout to arrange button for the digits and for the +, -, *, % operations. Add a text field to display the result		

10	<p>a. Write a Java program for handling mouse events.</p> <p>b. Write a Java program for handling key events</p>		
11	<p>1. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields Num1 and Num 2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception and display the exception in a message dialog box.</p>		
12	<p>a. Write a java program that simulates traffic light. The program lets the user select one of three lights: red, yellow or green. When a radio button is selected, the light is turned on, and only one light can be on at a time No 1 Light is on when the program starts.</p> <p>b. Write a Java program that allows the user to draw lines, rectangles and ovals</p>		
13	<p>1. Create a table in Table.txt file such that the first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using JTable component.</p>		

1. Vision, Mission & PEOs

VISION:

To meet the emerging trends in Computer Science and Engineering, strive for self-reliance enabled through high end research by adopting a futuristic approach.

MISSION:

M1: To impart qualitative education, prepare students refurbish their latent talents and aspire for a pragmatic career in Computer Science and Engineering.

M2: To provide an ambiance to develop strategic areas of advance study with perception to foster industry centric education in Computer Science and Engineering.

M3: To inculcate self-learning among students to make them self-reliant and socially responsible

Program Educational Objectives (PEOs):	
PEO1	To emphasize analytical, computational and programming based tools and methods of Computer Science and Engineering to solve real world problems.
PEO2	To mould the students careers through steering their confidence levels for better understanding, strengthening technical outlook for innovation and better communication at the job place where they are employed.
PEO3	To hone the technical skills for creation and productivity in Computer Science and Engineering beyond class room learning.
PEO4	To promote multi-disciplinary awareness through exposure to areas of project development and industrial training for sustainable competition in Research and Development.

2. POs & PSOs

PROGRAMME OUTCOMES (POs):

PO1.	ENGINEERING KNOWLEDGE: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2.	PROBLEM ANALYSIS: Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3.	DESIGN/DEVELOPMENT OF SOLUTIONS: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
PO4.	CONDUCT INVESTIGATIONS OF COMPLEX PROBLEMS: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5.	MODERN TOOL USAGE: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6.	THE ENGINEER AND SOCIETY: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
PO7.	ENVIRONMENT AND SUSTAINABILITY: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8.	ETHICS: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9.	INDIVIDUAL AND TEAM WORK: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10.	COMMUNICATION: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, give and receive clear instructions.
PO11.	PROJECT MANAGEMENT AND FINANCE: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12.	LIFE-LONG LEARNING: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OBJECTIVES (PSOs):

PSO1	Ability to design and develop computing system using mathematical knowledge and expertise from other disciplines.
PSO2	Ability to test and analyse quality of various systems to integrate them in larger computer systems.

3. Course objectives & Course outcomes

Course objectives:

The Student will:

1. Write java programs using arithmetic operators, control statements, type conversion, constructors and string handling
2. Explain how to write java programs using inheritance and polymorphism
3. Explain how to write java programs for creation of user defined packages and interfaces
4. Familiar with exception handling, multithreading and event handling

Explain how to write java programs using applets.

Course outcomes:

The Student will be able to:

1. Implement arithmetic operators, control statements, type conversion, constructors and string handling.
2. Apply the OOP concepts using java.
3. Create user defined Packages and Interfaces
4. Implement Exception handling and Multithreading
5. Design GUI using Applets and AWT.

4. Course Syllabus / List of Experiments (Syllabus Copy)

J.B.INSTITUTE OF ENGINEERING & TECHNOLOGY

UGC AUTONOMOUS

Bhaskar Nagar, Moinabad (M), RR Dist, Telangana-500075

B.Tech. CSE	L	T-P-D	C
II Year - II Semester	0	0-4-0	2

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

(Common to CSE, IT & ECM)

Course objectives:

The Student will :

Write java programs using arithmetic operators ,control statements, type conversion, constructors and string handling

Explain how to write java programs using inheritance and polymorphism

Explain how to write java programs for creation of user defined packages and interfaces

Familiar with exception handling, multithreading and event handling

Explain how to write java programs using applets.

Experiment 1:

Write java programs that implement the following

- a) Constructor b) Parameterized constructor
- c) Method overloading d) Constructor overloading.

Experiment 2:

a) Write a Java program that checks whether a given string is a palindrome or not.

Ex: MADAM is a palindrome.

b) Write a Java program for sorting a given list of names in ascending order.

c) Write a Java Program that reads a line of integers, and then displays each integer and the sum of all the integers (Use String Tokenizer class of java.util

Experiment 3:

Write java programs that uses the following keywords

- a) this b) super
- c) static d) final

Experiment 4:

a) Write a java program to implement method overriding

b) Write a java program to implement dynamic method dispatch.

c) Write a Java program to implement multiple inheritance.

d) Write a java program that uses access specifiers

Experiment 5:

a) Write a Java program that reads a file name from the user, then displays information about

whether the file exists, whether the file is readable, whether the file is writable, the type of file

and the length of the file in bytes.

b) Write a Java program that reads a file and displays the file on the screen, with a line number

before each line.

c) Write a Java program that displays the number of characters, lines and words in a text file

Experiment 6:

a) Write a Java program for handling Checked Exceptions.

b) Write a Java program for handling Unchecked Exceptions

Experiment 7:

a) Write a Java program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds.

b) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication

Experiment 8:

a) Develop an applet that displays a simple message.

b) Develop an applet that receives an integer in one text field, and computes its factorial value and returns it in another text field, when the button named “Compute” is clicked

Experiment 9:

Write a Java program that works as a simple calculator. Use a grid layout to arrange button for the digits and for the +, -, *, % operations. Add a text field to display the result

Experiment 10:

a) Write a Java program for handling mouse events.

b) Write a Java program for handling key events.

Experiment 11:

1. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields Num1 and Num 2.

The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception and display the exception in a message dialog box.

Experiment 12:

a) Write a java program that simulates traffic light. The program lets the user select one of three lights: red, yellow or green. When a radio button is selected, the light is turned on, and only one light can be on at a time No 1 Light is on when the program starts.

b) Write a Java program that allows the user to draw lines, rectangles and ovals

Experiment 13:

Create a table in Table.txt file such that the first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using JTable component

TEXT BOOKS:

1. Java;the complete reference,8th editon ,Herbert Schildt, TMH.
2. Java How to Program, Sixth Edition, H.M.Dietel and P.J.Dietel, Pearson Education/PHI.
3. Introduction to Java programming, Sixth edition, Y.Daniel Liang, Pearson Education.
4. Big Java, 2nd edition, Cay Horstmann, Wiley Student Edition, Wiley India Private Limited

Course outcomes:**The Student will be able to:**

6. Implement arithmetic operators ,control statements, type conversion, constructors and string handling .
7. Apply the OOP concepts using java.
8. Create user defined Packages and Interfaces
9. Implement Exception handling and Multithreading
10. Design GUI using Applets and AWT.

5. COs & POs Mapping

CO-PO/PSO Mapping Chart
(3/2/1 indicates strength of correlation)

3 – Strong; 2 – Medium; 1 - Weak

Course Outcomes (COs)	Program Outcomes (POs)												Program Specific Outcomes	
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO1	-	-	3	-	3	-	-	-	-	-	3	-	-	2
CO2	-	-	2	-	2	-	-	-	-	-	3	-	-	2
CO3	-	-	2	-	3	-	-	-	-	-	2	-	-	3
CO4	-	-	2	-	3	-	-	-	-	-	2	-	-	3
CO5	-	-	3	-	3	-	-	-	-	-	2	-	-	3
Average	-	-	2.40	-	2.80	-	-	-	-	-	2.40	-	-	2.60

6. Lab Manual

1 EXPERIMENT -I

Write a java program to implement the following:

- a) Constructor
- b) parameterized constructor
- c) Method overloading
- d) Constructor overloading

1 a) Constructor

```
public class MyDefaultConstructor
{

public MyDefaultConstructor(){
System.out.println("I am inside default constructor...");
}

public static void main(String a[])
{
MyDefaultConstructor mdc = new MyDefaultConstructor();
}
}
```

Output

I am inside default constructor...

1 b) Parameterized constructor

```
public class MyParameterizedConstructor
{
private String name;
public MyParameterizedConstructor(String str)
{
this.name = str;
System.out.println("I am inside parameterized constructor.");
System.out.println("The parameter value is: "+str);
}
public static void main(String a[])
{
MyParameterizedConstructor mpc = new
MyParameterizedConstructor("HIMAGIRI");
}
}
```

Output:

I am inside parameterized constructor.The
parameter value is: HIMAGIRI

1 c) constructor overloading

```
class DisplayOverloading
{
public void disp(char c)
{
System.out.println(c);
}
public void disp(char c, int num)
{
System.out.println(c + " "+num);
}
}
class Sample
{
public static void main(String args[])
{
DisplayOverloading obj = new DisplayOverloading();obj.disp('a');
obj.disp('a',10);
}
}
```

Output:

a
a 10

1 d) Constructor overloading

```
public class StudentData
{
private int stuID; private String
stuName;private int stuAge;
StudentData()
{
//Default constructorstuID = 100;
stuName = "srikanth";stuAge = 18;
}
StudentData(int num1, String str, int num2)
{
//Parameterized constructorstuID =
num1;
stuName = str; stuAge = num2;
}
//Getter and setter methodspublic int
getStuID() { return stuID;
}
public void setStuID(int stuID) {this.stuID =
stuID;
}
public String getStuName() {return
stuName;
}
public void setStuName(String stuName) {this.stuName
= stuName;
}
public int getStuAge() {return
stuAge;
}
public void setStuAge(int stuAge) {this.stuAge =
stuAge;
}
}
class TestOverloading
{
public static void main(String args[])
{
//This object creation would call the default constructor StudentData myobj
= new StudentData(); System.out.println("Student Name is:
"+myobj.getStuName());System.out.println("Student Age is:
"+myobj.getStuAge()); System.out.println("Student ID is:
"+myobj.getStuID());
}
```

```

/*This object creation would call the parameterized
* constructor StudentData(int, String, int)*/
StudentData myobj2 = new StudentData(555, "renuka", 25);
System.out.println("Student Name is: "+myobj2.getStuName());
System.out.println("Student Age is: "+myobj2.getStuAge());
System.out.println("Student ID is: "+myobj2.getStuID());
}
}

```

Output:

```

Student Name is: srikanthStudent Age
is: 18 Student ID is: 100 Student Name
is: renuka Student Age is: 25 Student
ID is: 555

```

EXPERIMENT- II:

2. a)write a java program that checks whether a given string is a palindrome or not

Ex: MADAM is a palindrome

```

import java.util.*;
class Palindrome
{
    public static void main(String args[])
    {
        String original, reverse = "";
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a string to check if it is a palindrome");original =
in.nextLine();
        int length = original.length();
        for ( int i = length - 1; i >= 0; i-- ) reverse =
reverse + original.charAt(i);if
(original.equals(reverse))
System.out.println("Entered string is a palindrome.");
        else
System.out.println("Entered string is not a palindrome.");=
    }
}

```

OUTPUT :

```

Enter a string to check if it is a palindromeMADAM
Entered string is a palindrome
-----

```


2.b)write a java program that reads a line of integers and then displays each integer and the sum of all the integers (use string tokenizer class of java.util).

```
import java.util.StringTokenizer;
import java.util.Scanner;
class tokens
{
public static void main(String[] args)
{
Scanner input=new Scanner(System.in);
String sentence=input.nextLine();
String temp;
int k,total=0;
StringTokenizer s1=new StringTokenizer(sentence);
System.out.println("Total Number of tokens:"+s1.countTokens());
while(s1.hasMoreTokens())
{
temp=s1.nextToken();
k=Integer.parseInt(temp);
total+=k;
System.out.print(k+"\t");
}
System.out.println("Sum of tokens :"+total);
}
}
```

Output:

12 43 78 98

Total Number of tokens:412 43 78 98

Sum of tokens : 231123 456 798

Total number of tokens:3123 456 798

Sum of tokens:1377

EXPERIMENT-III :

Write java programs that uses the following keywords

**a)this b)super
d)static d)final**

3 a) this

```
class Student
{
int rollno; String name;
float fee;
Student(int rollno,String name,float fee){
this.rollno=rollno;
this.name=name;this.fee=fee;
}
void display(){System.out.println(rollno+" "+name+" "+fee);}
}
class TestThis2
{
public static void main(String args[])
{
Student s1=new Student(111,"ankit",5000f); Student
s2=new Student(112,"sumit",6000f);s1.display();
s2.display();
}}
```

OUTPUT

111 ankit 5000
112 sumit 6000

3 b)SUPER

```
class Animal
{
String color="white";
}
class Dog extends Animal
{
String color="black";void
printColor(){
System.out.println(color);//prints color of Dog class System.out.println(super.color);//prints color
of Animal class
}
}
class TestSuper1
{
public static void main(String args[])
{
Dog d=new Dog();
d.printColor();
}}
```

Output:

blackwhite

--- 3 c)STATIC

```
class Student8{int rollno;
String name;
static String college ="ITS";

Student8(int r,String n){rollno = r;
name = n;
}
void display (){System.out.println(rollno+" "+name+" "+college);}

public static void main(String args[]){ Student8 s1 = new
Student8(111,"reuka"); Student8 s2 = new
Student8(222,"srikanth");

s1.display();
s2.display();
}
}
```

Output:

111 reuka ITS
222 srikanth ITS

4 d)FINAL

```
class Bike10
{
    final int speedlimit; //blank final variableBike10()
}
    speedlimit=70;
    System.out.println(speedlimit);
}
    public static void main(String args[]){
new Bike10();
}
}
```

OUTPUT : 70

EXPERIMENT-IV

4.a) write a java program to implement method overriding

```
class Vehicle
{
void run()
{
System.out.println("Vehicle is running");
}
}
class Bike2 extends Vehicle{
void run(){System.out.println("Bike is running safely");}

public static void main(String args[])
{
Bike2 obj = new Bike2();
obj.run();
}
```

Output: Bike is running safely

4.b) write a java program to implement dynamic method dispatch

```
class Animal{
void eat(){System.out.println("animal is eating...");}
}

class Dog extends Animal{
void eat(){System.out.println("dog is eating...");}

public static void main(String args[]){ Animal
a=new Dog();
a.eat();
}
}
```

Output: dog is eating...

4.c) write a java program to implement multiple inheritance

```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class BabyDog extends Dog{
void weep(){System.out.println("weeping...");}
}
class TestInheritance2{
public static void main(String args[]){BabyDog
d=new BabyDog(); d.weep();
d.bark();
d.eat();
}}
```

Output:

```
weeping...barking...
eating...
```

4.d) write a java program that uses access specifiers

```
;
public class A{
protected void msg(){System.out.println("Hello");}
}

//save by B.java package
mypack;import pack.*;

class B extends A{
public static void main(String args[]){B obj = new
B();
obj.msg();
}
}
```

Output : Hello

EXPERIMENT –V

5.a) write a java program that reads a file name from the user ,file name from the user,then displays information about whether the file exists,whether the file is readable,whether the file is writable,the type of file and the length of the file in bytes

```
import java.util.Scanner;
import java.io.File;
class FileDemo
{
public static void main(String[] args)
{
Scanner input=new Scanner(System.in);
String s=input.nextLine();
File f1=new File(s);
System.out.println("File Name:"+f1.getName());
System.out.println("Path:"+f1.getPath());
System.out.println("Abs Path:"+f1.getAbsolutePath());
System.out.println("Parent:"+f1.getParent());
System.out.println("This file is:"+(f1.exists()?"Exists":"Does not exists"));
System.out.println("Is file:"+f1.isFile());
System.out.println("Is Directory:"+f1.isDirectory());
System.out.println("Is Readable:"+f1.canRead());
System.out.println("Is Writable:"+f1.canWrite());
System.out.println("Is Absolute:"+f1.isAbsolute());
System.out.println("File Last Modified:"+f1.lastModified());
System.out.println("File Size:"+f1.length()+"bytes");
System.out.println("Is Hidden:"+f1.isHidden());
}
}
```

Output:

```
Fibonacci.java
File Name:Fibonacci.javaPath:
Fibonacci.java
Abs Path: c:\sameer\Fibonacci.javaParent: Null
This file is:ExistsIs file:true
Is Directory:falseIs
Readable:true Is Writable:true
Is Absolute:false
File Last Modified:1206324301937File Size:
406 bytes
Is Hidden:false
```

5.b) write a java program that reads a file and displays the file on the screen with a line number before each line

```
import java.util.*;import
java.io.*;

public class Test {
public static void main(String[] args) throws IOException
{
Scanner keyboard = new Scanner(System.in);
System.out.print("Enter a file name: "); String filename =
keyboard.nextLine();

File file = new File(filename); Scanner inputFile =
new Scanner(file);String line =
inputFile.nextLine();
int lineNumber=0;
while (inputFile.hasNext())
{
String name = inputFile.nextLine();
System.out.println(lineNumber+ ":"+name);`
linenumber++;
System.out.println(name);
}

inputFile.close();

}
```

} OUTPUT:

```
Enter a file name:a.txt
0 : abc
1: abcdefghij
2: 122345667899900
```


5.c) write a java program that displays the number of charecters,lines and wordsin a text file.

```
import java.io.*;class
wordcount
{
public static int words=0;public static
int lines=0; public static int chars=0;
public static void wc(InputStreamReader isr)throws IOException
{
int c=0;
boolean lastwhite=true;
while((c=isr.read())!=-1)
{
chars++; if(c=='\n')
lines++;
if(c=='\t' || c==' ' || c=='\n')
++words; if(chars!=0)
++chars;
} }
public static void main(String[] args)
{
FileReader fr;try
{
if(args.length==0)
{
wc(new InputStreamReader(System.in));
}
else
{
for(int i=0;i<args.length;i++)
{
fr=new FileReader(args[i]);wc(fr);
} }
}
catch(IOException ie)
{
return;
}System.out.println(lines+" "+words+" "+chars);
}}
```

Output:

My name is Renuka1 4 16

EXPERIMENT-V1

6.a) write a java program for handling checked exceptions

```
import java.io.*;class
Example {
public static void main(String args[]) throws IOException
{
FileInputStream fis = null;
fis = new FileInputStream("B:/myfile.txt");int k;

while(( k = fis.read() ) != -1)
{
System.out.print((char)k);
}
fis.close();
}
}
```

Output:

File content is displayed on the screen.

6.b)write a jav program for handling unchecked exceptions

```
class Example {
public static void main(String args[])
{
int num1=10;int num2=0;
/*Since I'm dividing an integer with 0
* it should throw ArithmeticException*/int
res=num1/num2; System.out.println(res);
}
}
```

OUTPUT:

Divide by Zero Exception: ArithmeticException

Note:

If you compile this code, it would compile successfully however when you will run it, it would throw ArithmeticException. That clearly shows that unchecked exceptions are not checked at compile-time, they are being checked at runtime.

EXPERIMENT-VII

7.a) write a java program that creates three threads.first thread displays “good morning” every one second,the second thread displays “hello” evry two seconds and the thread displays “welcome “ every three seconds.

```
class A extends Thread
{
synchronized public void run()
{
try
{
while(true)
{
sleep(1000); System.out.println("good morning");
}
}
catch(Exception e)
{ }
}
}
class B extends Thread
{
synchronized public void run()
{
try
{
while(true)
{
sleep(2000); System.out.println("hello");
}
}
catch(Exception e)
{ }
}
}
class C extends Thread
{
synchronized public void run()
{
try
{
while(true)
{
sleep(3000); System.out.println("welcome");
}
}
}
```

```
catch(Exception e)
{ }
}
}
class ThreadDemo
{
public static void main(String args[])
{
A t1=new A(); B t2=new
B();C t3=new C();t1.start();
t2.start();
t3.start();
}
}
```

Output:

```
E:\javamani>java ThreadDemogood
morning
good morninghello
good morningwelcome
good morninghello
good morningwelcome
hello
good morning good
morninghello
good morningwelcome
good morninghello
good morninggood
morninghello welcome
good morning
```

7.b) write a java program that correctly implements producer consumer problem using the concept of inter thread communication

```
class Q
{
int n;
boolean valueSet=false;
synchronized int get()
{
if(!valueSet)
try
{
wait();
}
catch(InterruptedException e)
{
System.out.println("Interrupted Exception caught");
}
System.out.println("Got:"+n);
valueSet=false;
notify();
return n;
}
synchronized void put(int n)
{
if(valueSet)
try
{
wait();
}
catch(InterruptedException e)
{
System.out.println("Interrupted Exception caught");
}
this.n=n;
valueSet=true;
System.out.println("Put:"+n);
notify();
}
}
class Producer implements Runnable
{
Q q;
Producer(Q q)
{
this.q=q;
new Thread(this,"Producer").start();
}
public void run()
{
```

```

int i=0;
while(true)
{
q.put(i++);
}
}
}
class Consumer implements Runnable
{
Q q;
Consumer(Q q)
{
this.q=q;
new Thread(this, "Consumer").start();
}
public void run()
{
while(true)
{
q.get();
}
}
}
class ProdCons
{
public static void main(String[] args)
{
Q q=new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-c to stop");
}
}

```

Output:Put:1 Got:1
Put:2 Got:2 Put:3
Got:3 Put:4 Got:4
Put:5 Got:5 Put:6
Got:6 Put:7 Got:7
Put:8 Got:8

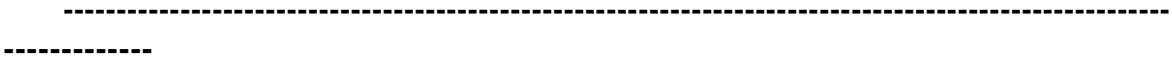
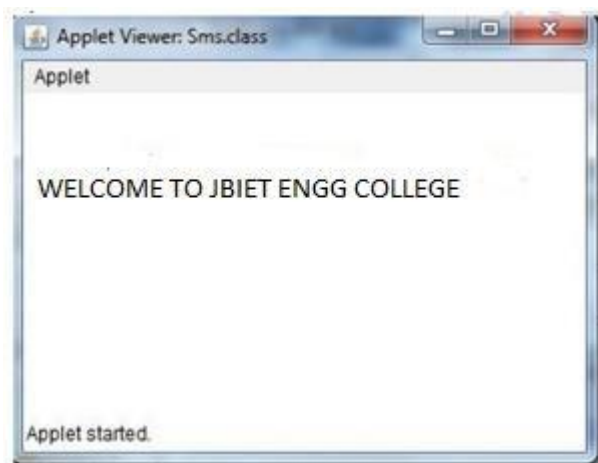
Put:9 Got:9 Put:10
Got:10Put:11Got:11
Put:12Got:12Put:13
Got:13Put:14Got:14

EXPERIMENT-VIII

8.a) Develop an applet in Java that displays a simple message.

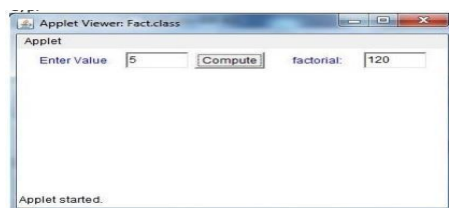
```
import java.applet.Applet;  
import java.awt.*;  
public class Sms extends Applet {  
    public void init() {  
    }  
    public void paint(Graphics g) {  
        g.setColor(Color.blue);  
        Font font = new Font("verdana", Font.BOLD, 15);  
        g.setFont(font);  
        g.drawString("WELCOME TO JBIET ENGG COLLEGE", 50, 50);  
    }  
}
```

Output:



8.b) Develop an applet in Java that receives an integer in one text field, and computes its factorial value and returns it in another text field, when the button named “Compute” is clicked

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class Fact extends Applet implements ActionListener {
    Label l1, l2, l3;
    TextField tf1, tf2;
    Button b1;
    public void init() {
        setSize(400, 200);
        FlowLayout g = new FlowLayout();
        setLayout(g);
        l1 = new Label("Enter Value");
        l1.setForeground(Color.BLUE);
        add(l1);
        tf1 = new TextField(5);
        tf1.setText("0");
        add(tf1);
        b1 = new Button("Compute");
        b1.addActionListener(this);
        add(b1);
        l3 = new Label();
        add(l3);
        l2 = new Label("factorial: ");
        l2.setForeground(Color.BLUE);
        add(l2);
        tf2 = new TextField(5);
        add(tf2);
    }
    public void actionPerformed(ActionEvent ae) {
        long n = Integer.parseInt(tf1.getText());
        long f = 1;
        while (n != 0) {
            f = f * n;
            n--;
        }
        tf2.setText(String.valueOf(f));
    }
}
```



Output:

EXPERMINT-1X

9) Write a Java program that works as simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, *, % operators. Add a text field to display the result. Handle any possible exceptions like divided by zero

```
package calc;
import javax.swing.*; import
javax.swing.event.*; import java.awt.*;
import java.awt.event.*;
class A extends JFrame implements ActionListener {
public JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, b13, b14, b15, b16; public
JTextField tf1;
public JPanel p; public String v =
""; public String v1 = "0"; public
String op = ""; public A() {
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); setSize(400,
400);
p = new JPanel(new FlowLayout()); tf1 = new
JTextField(10); p.add(tf1);
add(p);
setLayout(new GridLayout(0, 3)); b1 = new
JButton("1"); b1.addActionListener(this);
add(b1);
b2 = new JButton("2");
b2.addActionListener(this); add(b2);
b3 = new JButton("3");
b3.addActionListener(this); add(b3);
b4 = new JButton("4");
b4.addActionListener(this); add(b4);
b5 = new JButton("5");
b5.addActionListener(this); add(b5);
b6 = new JButton("6");
b6.addActionListener(this); add(b6);
b7 = new JButton("7");
b7.addActionListener(this); add(b7);
b8 = new JButton("8");
b8.addActionListener(this); add(b8);
```

```
b9 = new JButton("9");
b9.addActionListener(this);add(b9);
b10 = new JButton("0");
b10.addActionListener(this);add(b10);
b11 = new JButton("+");
b11.addActionListener(this);add(b11);
b12 = new JButton("-");
b12.addActionListener(this);add(b12);
b13 = new JButton("*");
b13.addActionListener(this);add(b13);
b14 = new JButton("/");
b14.addActionListener(this);add(b14);
b16 = new JButton("%");
b16.addActionListener(this);add(b16);
b15 = new JButton("=");
b15.addActionListener(this);add(b15);
setVisible(true);
}
public void actionPerformed(ActionEvent ae) {String b =
ae.getActionCommand();
switch (b) {case "1": {
v = v + "1";
tf1.setText(v);
}
break; case "2": {
v = v + "2";
tf1.setText(v);
}
break; case "3": {
v = v + "3";
tf1.setText(v);
}
break; case "4": {
v = v + "4";
tf1.setText(v);
}
break; case "5": {
```

```
v = v + "5";
tf1.setText(v);
}
break; case "6": {
v = v + "6";
tf1.setText(v);
}
break; case "7": {
v = v + "7";
tf1.setText(v);
}
break; case "8": {
v = v + "8";
tf1.setText(v);
}
break; case "9": {
v = v + "9";
tf1.setText(v);
}
break; case "0": {
v = v + "0";
tf1.setText(v);
}
break; case "+": {
op = "+";
v1 = tf1.getText();v = "";
}
break; case "-": {
op = "-";
v1 = tf1.getText();v = "";
}
break; case "*": {
op = "*";
v1 = tf1.getText();v = "";
}
break; case "/": {
op = "/";
v1 = tf1.getText();
```

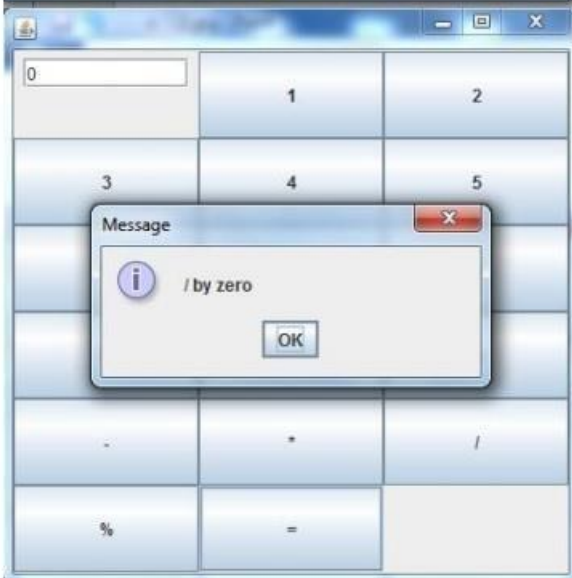
```
v = "";
}
break;
case "%": {
op = "%";
v1 = tf1.getText();v = "";
}
break; case "=": {
switch (op) {case "+": {
v = tf1.getText(); if
(v.equals("")) {v = "0";
}
long i = Long.parseLong(v1) + Long.parseLong(v);
tf1.setText(String.valueOf(i));
v="";
}
break; case "-": {
v = tf1.getText(); if
(v.equals("")) {v = "0";
}
long i = Long.parseLong(v1) - Long.parseLong(v);
tf1.setText(String.valueOf(i));
v="";
}
break; case "*": {
v = tf1.getText(); if
(v.equals("")) {v = "0";
}
long i = Long.parseLong(v1) * Long.parseLong(v);
tf1.setText(String.valueOf(i));
v="";
}
break; case "/": {try {
v = tf1.getText(); if
(v.equals("")) {v = "0";
}
long i = Long.parseLong(v1) / Long.parseLong(v);
tf1.setText(String.valueOf(i));
v="";
```

```

} catch (Exception ex) { JOptionPane.showMessageDialog(this,
ex.getMessage());
}
}
break;
case "%": { try {
v = tf1.getText(); if
(v.equals("")) { v = "0";
}
long i = Long.parseLong(v1) % Long.parseLong(v);
tf1.setText(String.valueOf(i));
v="";
} catch (Exception ex) { JOptionPane.showMessageDialog(this,
ex.getMessage());
}
}
break;
}
}
break;
}
}
}
public class Calc {
public static void main(String[] args) { A a = new
A();
}
}
}

```

Output:

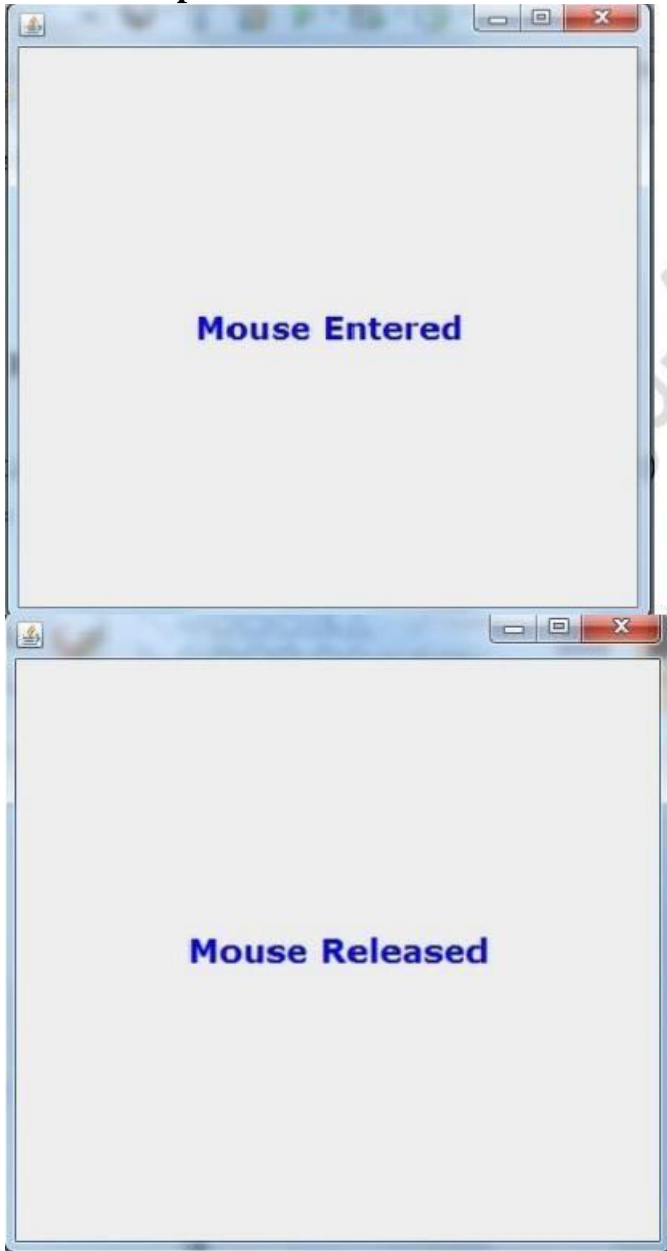


EXPERIMENT-X

10 a) Write a Java program that handles all mouse events and shows the eventname at the center of the window when a mouse event is fixed (Use Adapter Classes).

```
import javax.swing.*;import
java.awt.*;
import javax.swing.event.*;import
java.awt.event.*;
class A extends JFrame implements MouseListener {JLabel l1;
public A() { setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(400, 400);
setLayout(new GridBagLayout());l1 = new
JLabel();
Font f = new Font("Verdana", Font.BOLD, 20);l1.setFont(f);
l1.setForeground(Color.BLUE);
l1.setAlignmentX(Component.CENTER_ALIGNMENT);
l1.setAlignmentY(Component.CENTER_ALIGNMENT);add(l1);
addMouseListener(this);
setVisible(true);
}
public void mouseExited(MouseEvent m) {
l1.setText("Mouse Exited");
}
public void mouseEntered(MouseEvent m) {
l1.setText("Mouse Entered");
}
public void mouseReleased(MouseEvent m) {
l1.setText("Mouse Released");
}
public void mousePressed(MouseEvent m) {
l1.setText("Mouse Pressed");
}
public void mouseClicked(MouseEvent m) {
l1.setText("Mouse Clicked");
}
}
public class Mevents {
public static void main(String[] args) {A a = new
A();
}
}
```

Output:

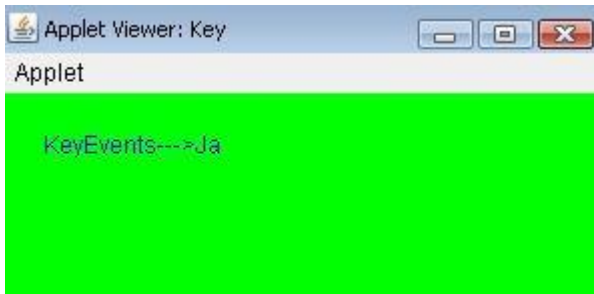


10.b) write a java program for handling key

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
<applet code="Key" width=300 height=400>
</applet>
*/
public class Key extends Applet
implements KeyListener
{
int X=20,Y=30;
String msg="KeyEvents--->";
public void init()
{
addKeyListener(this);
requestFocus();
setBackground(Color.green);
setForeground(Color.blue);
}
public void keyPressed(KeyEvent k)
{
showStatus("KeyDown");
int key=k.getKeyCode();
switch(key)
{
case KeyEvent.VK_UP:
showStatus("Move to Up");
break;
case KeyEvent.VK_DOWN:
showStatus("Move to Down");
break;
case KeyEvent.VK_LEFT:
showStatus("Move to Left");
break;
case KeyEvent.VK_RIGHT:
showStatus("Move to Right");
break;
}
repaint();
}
public void keyReleased(KeyEvent k)
{
showStatus("Key Up");
}
public void keyTyped(KeyEvent k)
{
msg+=k.getKeyChar();
repaint();
}
```

```
}  
public void paint(Graphics g)  
{  
    g.drawString(msg,X,Y);  
}  
}
```

Output:



]

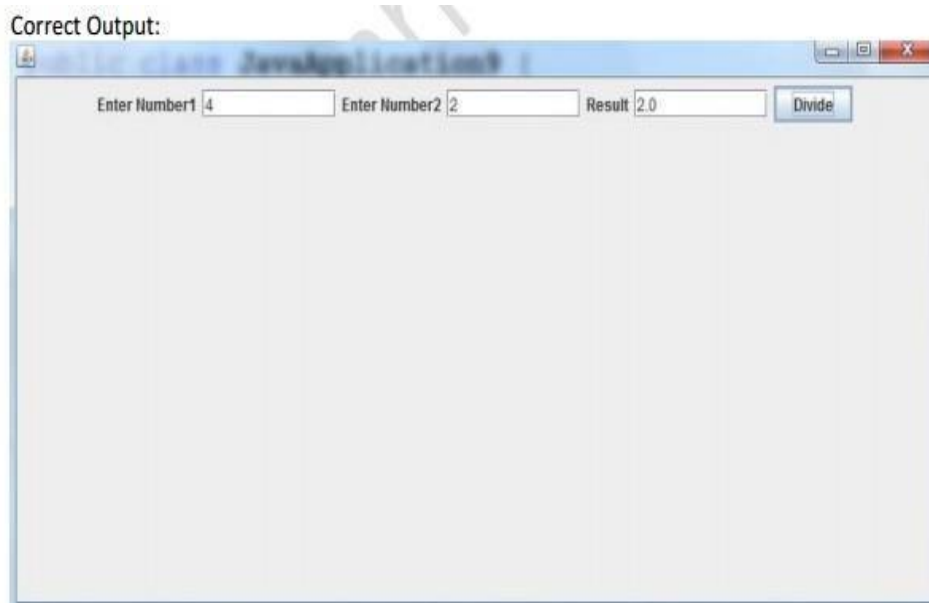
EXPERIMENT-XI

11) Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

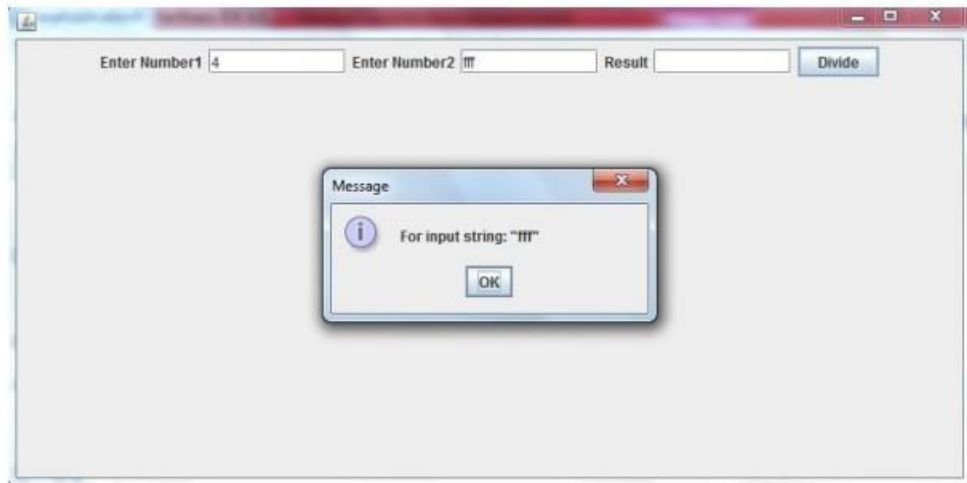
```
package javaapplication9;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
class A extends JFrame implements ActionListener {
    JLabel l1, l2, l3;
    JTextField tf1, tf2, tf3;
    JButton b1;
    A() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());
        l1 = new JLabel("Welcome");
        setSize(800, 400);
        l1 = new JLabel("Enter Number1");
        add(l1);
        tf1 = new JTextField(10);
        add(tf1);
        l2 = new JLabel("Enter Number2");
        add(l2);
        tf2 = new JTextField(10);
        add(tf2);
        l3 = new JLabel("Result");
        add(l3);
        tf3 = new JTextField(10);
        add(tf3);
        b1 = new JButton("Divide");
        add(b1);
        b1.addActionListener(this);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent ae) {
        try {
            int a = Integer.parseInt(tf1.getText());
            int b = Integer.parseInt(tf2.getText());
            if(b==0)
                throw new ArithmeticException(" Divide by Zero Error");
            float c = (float) a / b;
            tf3.setText(String.valueOf(c));
        } catch (NumberFormatException ex) {
```

```
JOptionPane.showMessageDialog(this, ex.getMessage());
} catch (ArithmeticException ex) {
JOptionPane.showMessageDialog(this, ex.getMessage());
}
}
}
public class JavaApplication9 {
public static void main(String[] args) {
A a = new A();
}
}
```

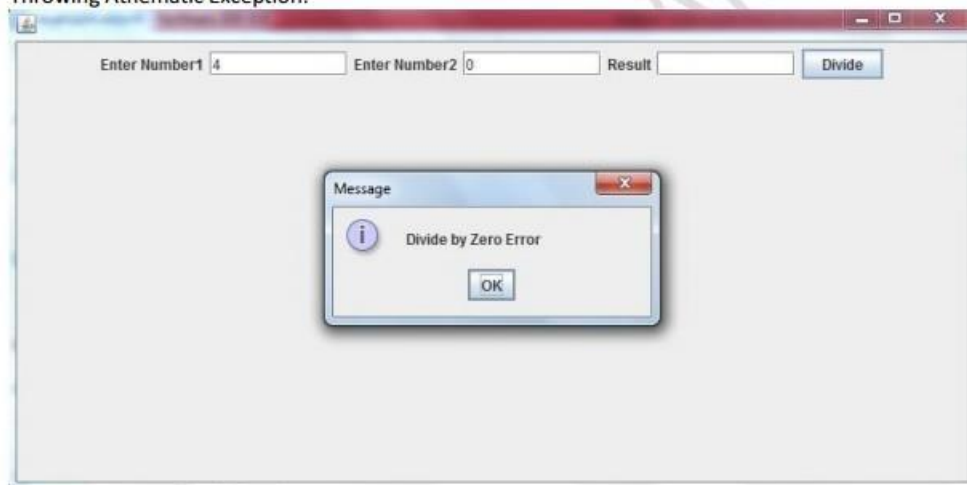
Output:



Throwing Number Format Exception:



Throwing Arithmetic Exception:



EXPERIMENT-XII

12 a) Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow or green with radio buttons. On selecting a button, an appropriate message with “Stop” or “Ready” or “Go” should appear above the buttons in selected color. Initially, there is no message shown.

```
import javax.swing.*; import
javax.swing.event.*; import java.awt.*;
import java.awt.event.*;
class A extends JFrame implements ItemListener { public JLabel
l1, l2;
public JRadioButton r1, r2, r3; public
ButtonGroup bg; public JPanel p, p1;
public A() { setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new GridLayout(2, 1));
setSize(800, 400);
p = new JPanel(new FlowLayout()); p1 = new
JPanel(new FlowLayout()); l1 = new JLabel();
Font f = new Font("Verdana", Font.BOLD, 60); l1.setFont(f);
add(l1);
p.add(l1);
add(p);
l2 = new JLabel("Select Lights"); p1.add(l2);
JRadioButton r1 = new JRadioButton("Red Light");
r1.setBackground(Color.red);
p1.add(r1); r1.addItemListener(this);
JRadioButton r2 = new JRadioButton("Yellow Light");
r2.setBackground(Color.YELLOW);
p1.add(r2); r2.addItemListener(this);
JRadioButton r3 = new JRadioButton("Green Light");
r3.setBackground(Color.GREEN);
p1.add(r3); r3.addItemListener(this);
add(p1);
bg = new ButtonGroup(); bg.add(r1);
bg.add(r2);
bg.add(r3); setVisible(true);
}
```

```
public void itemStateChanged(ItemEvent i) { JRadioButton
jb = (JRadioButton) i.getSource();switch (jb.getText()) {
case "Red Light": { l1.setText("STOP");
l1.setForeground(Color.red);
}
break;
case "Yellow Light": { l1.setText("Ready");
l1.setForeground(Color.YELLOW);
}
break;
case "Green Light": { l1.setText("GO");
l1.setForeground(Color.GREEN);
}
break;
}
}
}
}
public class TLights {
public static void main(String[] args) { A a = new
A();
}
}
```

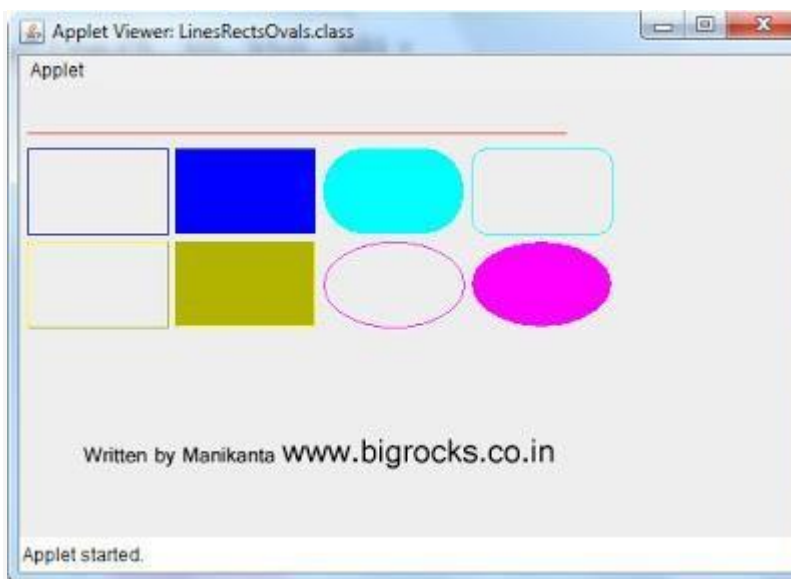
Output:



12 b) Write a Java program that allows the user to draw lines, rectangles and ovals.

```
import java.applet.*; import
java.awt.*; import javax.swing.*;
public class LinesRectsOvals extends JApplet
{
    public void paint(Graphics g)
    {
        super.paint(g);
        g.setColor(Color.red);
g.drawLine(5,30,350,30);
        g.setColor(Color.blue);
        g.drawRect(5,40,90,55);
        g.fillRect(100,40,90,55);
        g.setColor(Color.cyan);
        g.fillRoundRect(195,40,90,55,50,50);
        g.drawRoundRect(290,40,90,55,20,20);
        g.setColor(Color.yellow);
        g.draw3DRect(5,100,90,55,true);
        g.fill3DRect(100,100,90,55,false);
        g.setColor(Color.magenta);
        g.drawOval(195,100,90,55);
        g.fillOval(290,100,90,55);
    }
}
```

Output:



EXPERIMENT-XIII

Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using labels in Grid Layout.

```
package tbl; import java.io.*;
import java.util.*;import
java.awt.*;
import java.awt.event.*; import
javax.swing.*; import
javax.swing.event.*;class A extends
JFrame { public A() {
setSize(400, 400);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);GridLayout g
= new GridLayout(0, 3);
setLayout(g);try {
FileInputStream fin = new FileInputStream("D:\\emp.txt");Scanner sc =
new Scanner(fin).useDelimiter(",");
String[] arrayList;String a;
while (sc.hasNextLine()) {a =
sc.nextLine(); arrayList = a.split(",");
for (String i : arrayList) {add(new
JLabel(i));
}
}
} catch (Exception ex) {
}
setDefaultLookAndFeelDecorated(true);pack();
setVisible(true);
}
}
public class Tbl {
public static void main(String[] args) {A a = new
A();
}
}
```

Output:

eno,ename,mobile
101,RavikumarRanga,9849211983

102,Gurulingam,949459306

103,Gsr,9553122275



eno	ename	mobile
101	RavikumarRanga	9849211983
102	Gurulingam	949459306
103	Gsr	9553122275

