



**ALAGAPPA UNIVERSITY**  
(Accredited with 'A+' Grade by NAAC (with CGPA: 3.64) in the Third Cycle and Graded as  
category - I University by MHRD-UGC)  
(A State University Established by the Government of Tamilnadu)



**KARAIKUDI – 630 003**

**DIRECTORATE OF DISTANCE EDUCATION**

**B.Sc.[Information Technology]  
III Semester**

**Lab: Internet and Java Programming Lab**

Copy Right Reserved

For Private Use only

**Author:**  
**Mrs. G.Shanthi,**  
**System Programmer,**  
Department of Computer Applications,  
Alagappa University,  
Karaikudi.

**“The Copyright shall be vested with Alagappa University”**

**All rights reserved. No part of this publication which is material protected by this copyright notice may be reproduced or transmitted or utilized or stored in any form or by any means now known or hereinafter invented, electronic, digital or mechanical, including photocopying, scanning, recording or by any information storage or retrieval system, without prior written permission from the Alagappa University, Karaikudi, Tamil Nadu.**

**Reviewer,**  
**Dr. T. Meyyappan,**  
Professor of Computer Science,  
Alagappa University,  
Karaikudi.

## **Internet and Java Programming Lab**

<b>Unit No.</b>	<b>Syllabi</b>	<b>Mapping in Book</b>
<b>BLOCK 1: JAVA FUNDAMENTAL PROBLEMS</b>		
I	Simple Java Problems	1-5
II	Class and Objects Exercises	6-13
III	Conditional control Programs	13-15
IV	Looping Statement Exercises	15-18
<b>BLOCK 2: OOP CONCEPTS EXPERIMENTS</b>		
V	Polymorphism: Function overloading programs	19-22
VI	Operator Overloading , Inheritance programs	22-31
VII	Interface and Packages Programs	31-36
VIII	Message Passing programs	36-37
<b>BLOCK 3: THREAD &amp; VIRTUAL FUNCTION</b>		
IX	Threads Exercises	38-40
1X	Virtual function programs	40
<b>BLOCK 4: I/O AND EXCEPTION HANDLING</b>		
XI	I/O manipulation programs	41-43
XII	Exception handling programs	43-49
<b>BLOCK 5: NETWORK PROGRAMMING</b>		
XIII	Applet programs	50-57
XIV	Implementation of simple network programs	57-60
	Model Question Paper	61

---

# CHAPTER I INTRODUCTION

---

## 1.1 Overview

---

Java is a programming language from Sun Microsystems (Sun) developed by James Gosling in 1991. Java's goal is to write a program once and operate it on various operating systems. New upgraded versions of Java have been published over time. Java's latest version is Java 1.8, also known as Java 8.

Java is described by a specification and consists of a programming language, a compiler, key libraries and runtime (Java virtual machine). Java runtime enables software designers to write program code in languages other than the Java programming language that still operates on the virtual Java machine. The Java platform is normally connected to the Java virtual machine and the Java core libraries. The Java Runtime Environment (JRE) and the Java Development Kit (JDK) typically come in two types. The JRE consists of the libraries of the JVM and the Java class. These have the features needed to begin Java programs.

The JDK also includes the instruments for developing Java programs. The JDK therefore consists of a Java compiler, the Java virtual machine and the Java class libraries. As plain text document, Java source files are written. Usually the programmer writes Java source code for programming in an Integrated Development Environment (IDE). An IDE promotes the programmer in writing code tasks, e.g. providing source code auto-formatting, highlighting keywords, etc.

---

## 1.2 Software Requirements

---

We need to install a software program called Java SE Development Kit (or short JDK, and SE means Standard Edition) to write and run a Java program. In essence, a JDK includes JRE (Java Runtime Environment): is the heart of the Java platform that allows Java programs to be run on our computer. The JRE involves JVM (Java Virtual Machine) that operates and executes Java programs by translating from bytecode to platform-dependent code (Java programs are compiled into a bytecode intermediate form) and other key libraries such as collections, File I / O, networking, etc. It also includes tools and libraries to promote the growth of Java.

Java developer should be familiar with:

- **javac.exe**: is Java compiler that translates programs written in Java code into bytecode form.
- **java.exe**: is the Java Virtual Machine launcher that executes bytecode.

**NOTES**

## NOTES

**1.3 Simple Example:**

Step 1: Open a simple text editor program such as Notepad and type the following content:

```
public class HelloWorld {  
public static void main(String[] args) {  
System.out.println("Hello world!");  
}  
}
```

Step 2: Save the file as **HelloWorld.java** (note that the extension is **.java**) under a directory, let's say, **C:\jdk1.3\bin**

Step 3: Go to Command prompt. Type the following command to change the current directory to the one where the source file is stored:

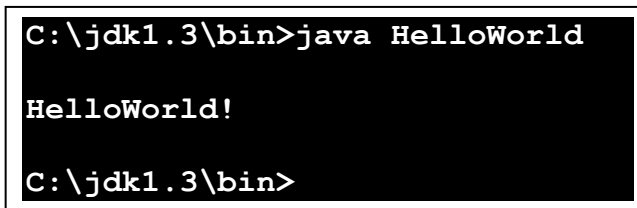
**cd C:\jdk1.3\bin**

And type the following command:

**javac HelloWorld.java**

Step 4: Type the following command:

**javaHelloWorld**



```
C:\jdk1.3\bin>java HelloWorld  
  
HelloWorld!  
  
C:\jdk1.3\bin>
```

---

## CHAPTER II

### BLOCK I: JAVA FUNDAMENTAL PROBLEMS

---

**NOTES****2.1 SIMPLE JAVA PROBLEMS:****1. Find the Greatest of three Numbers**

---

```
import java.io.*;
class Great
{
public static void main(String args[])
{
inta,b,c;
try
{
DataInputStream d= new DataInputStream(System.in);
System.out.println("enter the value of a:");
a=Integer.parseInt(d.readLine());
System.out.println("enter the value of b:");
b=Integer.parseInt(d.readLine());
System.out.println("enter the value of c:");
c=Integer.parseInt(d.readLine());
if(a>b)
{
System.out.println("greatest no is a="+a);
}
else if(c>b)
{
System.out.println("greatest no is c="+c);
}
else
{
System.out.println("greatest no is b="+b);
}}
}
```

## NOTES

```
catch (Exception e)
```

```
{
}}}
```

**Output:**

```
C:\jdk1.3\bin>javac Great.java
C:\jdk1.3\bin>java Great
enter value of a:
5
enter value of b:
9
enter value of c:
3
greatest no is b 9
```

---

**2. Write a Java program for Swapping of Two Numbers**

---

```
import java.io.*;
public class SwapNumbers
{
    public static void main(String[] args)
    {
        float a=1.20f,b=2.45f;
        System.out.println("--Before swap--");
        System.out.println("First number="+a);
        System.out.println("Second number="+b);
        float t=a; a=b;
        b=t;
        System.out.println("--After swap--");
        System.out.println("First number="+a);
        System.out.println("Second number="+b);
    }
}
```

**Output:**

```
C:\jdk1.3\bin>javac SwapNumbers.java
C:\jdk1.3\bin>java SwapNumbers
--Before swap--
First number =1.2
Second number = 2.4
--After swap--
First number = 2.4
Second number = 1.2
```

**NOTES****3. Fibonacci Series**

```
import java.io.*;
public class fibbonaci
{
public static void main(String[] args) {
int count = 7, num1 = 0, num2 = 1;
System.out.print("Fibonacci Series of "+count+" numbers:");
for (inti = 1; i<= count; ++i)
{
System.out.print(num1+" ");
int sum = num1 + num2;
num1 = num2;
num2 = sum;
}
}
```

**Output:**

```
C:\jdk1.3\bin>javac fib.java
C:\jdk1.3\bin>java fib
Fibonacci Series of 7 numbers: 0 1 1 2 3 5 8
```



---

## 2.2 Class and Objects:

---

### NOTES

---

### 4. Write a Java Program to define a class and its constructor

---

```
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
{
```

**NOTES**

```
public static void main(String args[])
{
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:\jdk1.3\bin>javacstudentdemo.java
```

```
c:\jdk1.3\bin>java studentdemo
```

```
raju 1234556 47 78
john 3426658 96 84
raju 1234556 47 78
```

**5. Program to practice using String class and its methods.**

```
Import java.lang.String;
class stringdemo
{
public static void main(String args[]) {
String s1=new String("alagappa university");
String s2=new String ("ALAGAPPA UNIVERSITY");
System.out.println(" The string s1 is : " +s1);
System.out.println(" The string s2 is : " +s2);
System.out.println(" Length of the string s1 is : " +s1.length());
System.out.println(" The first occurrence of u is at the position : "
+s1.indexOf('u'));
System.out.println(" The String in Upper Case : " +s1.toUpperCase());
```

**NOTES**

```
System.out.println(" The String in Lower Case : " +s2.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 " +s2);
else
System.out.println( s1 + " is smaller than " +s2);
System.out.println(" Character at an index of 6 is : " +s1.charAt(6));
String s3=s1.substring(5,8);
System.out.println(" Extracted substring is :"+s3);
System.out.println(" After Replacing a with g in s1 : " + s1.replace('a','g'));
String s4=" This is a book ";
System.out.println(" The string s4 is :"+s4);
System.out.println(" After trim()      :"+s4.trim());
}
}
```

**Output:**

```

c:\jdk1.3\bin>javacstringdemo.java

c:\jdk1.3\bin>java stringdemo

Thestrings1is :alagappa university
Thestrings2is :ALAGAPPA UNIVERSITY
Lengthofthestrings1is :19
Thefirstoccurrenceofuis attheposition:9
TheStringinUpperCase:ALAGAPPA UNIVERSITY
TheStringinLowerCase:alagappa university
s1equals tos2:false
s1equalsignorecasetos2:true
AftercompareTo()
alagappa university is greatherthanALAGAPPA UNIVERSITY
Characteratanindexof6is :p
Extractedsubstringis :appa
AfterReplacingawithgins1:glgggppg university
Thestrings4is:Thisis abook
After trim():Thisis abook

```

**NOTES****6. Program to practice using StringBuffer class and its methods.**

```

import java.lang.String;
class stringbufferdemo
{
public static void main(String args[])
{
StringBuffersb=new StringBuffer("This is my college");
System.out.println("This string sb is : " +sb);
System.out.println("The length of the string sb is : " +sb.length());
System.out.println("The capacity of the string sb is : " +sb.capacity());
System.out.println("The character at an index of 6 is : " +sb.charAt(6));
sb.setCharAt(3,'x');
System.out.println("After setting char x at position 3 : " +sb);
System.out.println("After appending : " +sb.append(" in karaikudi "));

```

**NOTES**

```

System.out.println("After inserting : " +sb.insert(19,"ALU "));
System.out.println("After deleting : " +sb.delete(19,22));
}
}

```

**Output:**

```

c:\jdk1.3\bin>javacstringbufferdemo.java
c:\jdk1.3\bin>java stringbufferdemo

This stringsbis :Thisismycollege
Thelengthofthestringsbis :18
Thecapacityofthestringsbis :34
Thecharacteratanindexof6is :s
Aftersettingcharatposition3:Thixismycollege
Afterappending:Thixismycollegeinkaraikudi
Afterinserting:Thixismycollege ALUinkaraikudi
Afterdeleting:Thixismycollegeinkaraikudi

```

---

**7. Program to implement Vector class and its methods.**

---

```

import java.lang.*;
import java.util.Vector;
import java.util.Enumeration;
class vectordemo
{
public static void main(String args[])
{
Vector v=new Vector(); v.addElement("one");
v.addElement("two"); v.addElement("three");
v.insertElementAt("zero",0); v.insertElementAt("oops",3);
v.insertElementAt("four",5);
System.out.println("Vector Size :"+v.size());
System.out.println("Vector apacity :"+v.capacity());

```

**NOTES**

```

System.out.println(" The elements of a vector are :");
Enumeration e=v.elements();
while(e.hasMoreElements())
System.out.println(e.nextElement() + " ");
System.out.println();
System.out.println("The first element is : " +v.firstElement());
System.out.println("The last element is : " +v.lastElement());
System.out.println("The object oops is found at position :
"+v.indexOf("oops"));
v.removeElement("oops");
v.removeElementAt(1);
System.out.println("After removing 2 elements ");
System.out.println("Vector Size :"+v.size());
System.out.println("The elements of vector are :");
for(inti=0;i<v.size();i++) System.out.println(v.elementAt(i)+" ");
} }

```

**Output:**

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

```
C:\jdk1.6.0_26\bin>java vectordemo
```

```
Vector Size :6
```

```
Vector Capacity :10
```

```
The elements of a vector are :
```

```
zero
```

```
one
```

```
two
```

```
oops
```

```
three
```

```
four
```

```
The first element is : zero
```

```
The last element is : four
```

```
The object oops is found at position: 3
```

```
After removing 2 elements
```

```
Vector Size :4
```

## NOTES

The elements of vector are :

zero  
two  
three  
four

### 8.ProgramtoimplementWrapperclasses andtheirmethods.

```
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,floatI,int y) {
int year=1; float sum=P;
while(year<=y) {
sum=sum+(P*I)/100; year++;
}
return sum;
}}

```

**Output:**

```

C:\jdk1.3\bin>javacwrapperdemo.java
C:\jdk1.3\bin>javawrapperdemo
ENTERTHE PRINCIPAL AMOUNT 1000
ENTERTHE INTERESTRATE 2
ENTERTHE NUMBEROF YEARS 1
FINAL VALUEIS:1020.0
E:\jdk1.3\bin>javawrapperdemo
ENTERTHE PRINCIPAL AMOUNT 1000
ENTERTHE INTERESTRATE 2
ENTERTHE NUMBEROF YEARS 2
FINAL VALUEIS:1040.0

```

**2.3 Conditional control in java****9. Program to implement array of objects.**

```

importjava.lang.*;
public class EmployeeTest {
public static void main(String args[]) {
Employee[] staff = new Employee[3];
staff[0] = new Employee("Ram", 3500);
staff[1] = new Employee("Sudha", 7500);
staff[2] = new Employee("Tony", 3800);
}
}

```

**NOTES**



**NOTES**

```

for (inti = 0; i < 3; i++)
    staff[i].print(); }
}
class Employee {
    private String name; private double salary;
    public Employee(String n, double s) {
        name = n; salary = s;
    }
    public void print() {
        System.out.println(name + " " + salary); }
}

```

**Output:**

```

C:\jdk1.3\bin>javacEmployeeTest.java
C:\jdk1.3\bin>javaEmployeeTest
Ram3500.0
Sudha7500.0
Tony 3800.0

```

---

**10. Program to demonstrate this pointer**

---

```

import java.lang.*;
class emp
{
    String name; int id;
    String address;
    void getdata(String name,int id,String address) {
        this.name=name; this.id=id; this.address=address;
    }
    void putdata() {

```

**NOTES**

```

System.out.println("Employee details are :");
System.out.println("Name :" +name);
System.out.println("ID :" +id);
System.out.println("Address :" +address);
} }
classempdemo {
public static void main(String args[]) {
emp e=new emp(); e.getdata("Raja",76859,"Alagappa"); e.putdata();} }

```

**Output:**

```

c:\jdk1.3\bin>javacempdemo.java
c:\jdk1.3\bin>java empdemo
Employeeedetails are:
Name :Raja
ID :76859
Address :Alagappa

```

---

**2.4 Looping using java****11. Program for alphabetical order**

---

```

import java.io.*;
class alpha
{
public static void main(String args[])throws IOException
{
String name[],t;
inti,j,x,n;
DataInputStream ds=new DataInputStream(System.in);
System.out.println("Enter the number of names");
n=Integer.parseInt(ds.readLine());
name=new String[n];
System.out.println("Enter the "+n+" names");

```

**NOTES**

```
for(i=0;i<n;i++)
{
name[i]=ds.readLine();
}
for(i=0;i<n-1;i++)
{
for(j=0;j<n-1-i;j++)
{
x=name[j].compareTo(name[j+1]);
if(x>0)
{
t=name[j];
name[j]=name[j+1];
name[j+1]=t;
}}}
System.out.println("ALPHABETICAL ORDER");
for(i=0;i<n;i++)
System.out.println(name[i]);
}}
```

**Output:**

```
C:\jdk1.3\bin>java alpha
Enter the number of names
3
Enter the 3 names
janaki
banu
sara
ALPHABETICAL ORDER
banu
janaki
sara
```

---

## 12. Sorting set of numbers

---

**NOTES**

```
import java.io.*;
class sort
{
public static void main(String args[])throws IOException
{
int a[];
int p,n,c,t,i;
DataInputStream ds=new DataInputStream(System.in);
System.out.println("Enter the number of elements");
n=Integer.parseInt(ds.readLine());
a=new int[n];
System.out.println("Enter the "+n+" number");
for(i=0;i<n;i++)
{
a[i]=Integer.parseInt(ds.readLine());
}
for(p=0;p<n-1;p++)
{
for(c=0;c<n-1-p;c++)
{if(a[c]>a[c+1])
{
t=a[c];
a[c]=a[c+1];
a[c+1]=t;
}}}
System.out.println("ASCENDING\tDESCENDING");
for(i=0;i<n;i++)
{
System.out.println(a[i]+"\\t\\t"+a[n-1-i]);
}}}
```

NOTES

**Output:**

```
C:\jdk1.3\bin>javac sort.java
C:\jdk1.3\bin>java sort
Enter the number of elements
5
Enter the 5 numbers
8
3
6
7
1
ASCENDING  DESCENDING
1          8
37
6          6
7          3
8          1
```

---

## CHAPTER III

---

Internet and Java Programming

NOTES

### BLOCK II: OOP CONCEPTS

#### 3.1 POLYMORPHISM – METHOD OVERLOADING

---

Polymorphism in Java is a concept by which we can perform a single action in different ways. There are two types of polymorphism in Java: compile-time polymorphism and runtime polymorphism. Method overloading is an example of static polymorphism, while method overriding is an example of dynamic polymorphism.

##### 13. Program for adding two numbers using method overloading and 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 args[]) {
add obj=new add();
obj.display(10,20);
obj.display(10.2,20.2);
}
}
```

*Self-Instructional Material*

**NOTES****Output:**

```
c:\jdk1.3\bin>javacadd_demo.java
```

```
c:\jdk1.3\bin>java add_demo
```

```
The sum of 10 & 20 is 30
```

```
The sum of 10.2 & 20.2 is 30.4
```

---

**14. Program to find the area of square and rectangle using method overloading**


---

```
import java.io.*;
class area
{
int x,y,z;
area()
{
x=y=z=0;
}
void area(int x1)
{
intsq=x1*x1;
System.out.println("area of square="+sq);
}
void area(int x2,int y2)
{
intrect=x2*y2;
System.out.println("area of rectangle="+rect);
}
}
class overloading
```

**NOTES**

```

{
public static void main(String args[])
{
area a=new area();
a.area(10);
a.area(10,20);
}}

```

**Output:**

```

C:\jdk1.3\bin>javac overloading.java

C:\jdk1.3\bin>java overloading

area of square=100

area of rectangle=200

```

**15.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 args[]) {

```



**NOTES**

```

outerobj=new outer();
outer.innerobj=outobj.new inner();
inobj.display();
}
}

```

**Output:**

```

C:\jdk1.3\bin>javacnesteddemo.java

C:\jdk1.3\bin>javanesteddemo

m = 10
n=20

```

***Note: Java does not support operator overloading***

**3.2 Inheritance****16. Program to demonstrate use of subclass**

```

import java.lang.*;
class parent
{
int m;
void get_m(int m)
{
this.m=m; }
void display_m()
{
System.out.println("This is from parent : m = " +m);
}}

class child extends parent {

```

**NOTES**

```
int n;
void get_n(int n)
{
    this.n=n; }

void display_n()
{
    System.out.println("This is from child : n = " +n); }
}

class childdemo
{
    public static void main(String args[])
    {
        child c=new child();
        c.get_m(10);
        c.get_n(20);
        c.display_m();
        c.display_n();
    } }
}
```

**Output:**

```
C:\jdk1.3\bin>javac chiddemo.java
```

```
C:\jdk1.3\bin>javachiddemo
```

```
This is from parent:m= 10
This is from child:n= 20
```

**NOTES****17. Program for banking process using class and objects**

```
import java.io.*;

class acc
{
    int acno, balance;

    void open_act(int a, int b)
    {
        acno = a;
        balance = b;
    }

    void credit(int amt)
    {
        balance = balance + amt;
        show_balance();
    }

    void withdraw(int amt)
    {
        if (balance - amt >= 100)
        {
            balance = balance - amt;
            show_balance();
        }
        else
            System.out.println("Sorry: No enough balance");
        show_balance();
    }

    void show_balance()
    {
        System.out.println("accno:" + acno);
    }
}
```

```
System.out.println("Current balance:"+balance);}}}
```

```
class bank
{
public static void main(String args[])throws IOException
{
acc a=new acc();
intano,amt,ch;
DataInputStream ds=new DataInputStream(System.in);
do
{
System.out.println("1.open new account");
System.out.println("2.credit in your account");
System.out.println("3.withdraw from your account");
System.out.println("4.Show balance");
System.out.println("Exit");
System.out.println("Enter your choice");
ch=Integer.parseInt(ds.readLine());
switch(ch)
{
case 1:
System.out.println("Enter your account number");
ano=Integer.parseInt(ds.readLine());
System.out.println("Enter the initial amount");
amt=Integer.parseInt(ds.readLine());
a.open_act(ano,amt);
break;
case 2:
System.out.println("Enter your credit amount");
amt=Integer.parseInt(ds.readLine());
a.credit(amt);
break;
```

## NOTES

**NOTES**

```

case 3:
System.out.println("Enter your withdraw amount");
amt=Integer.parseInt(ds.readLine());
a.withdraw(amt);
break;
case 4:
a.show_balance();
break;
case 5:
break;
}}
while(ch<5);
}}
```

**Output:**

```

C:\jdk1.3\bin>java bank
1.
  pen new account
2.
  redit in your account
3.
  ithdraw from your account
4.
  how balance
5.
  xit
Enter your choice
1
Enter the account number
2973
Enter the initial amount
20000
1.
  pen new account
2.
  redit in your account
3.
  ithdraw from your account
```

**NOTES**

```
4.  
    how balance  
5.  
    xit
```

```
Enter your choice  
2  
Enter your credit amount  
15000  
accno:2973  
Current balance:35000  
1.  
    pen new account  
2.  
    redit in your account  
3.  
    ithdraw from your account  
4.  
    how balance  
5.  
    xit  
Enter your choice  
3  
Enter your withdraw amount  
10000  
accno:2973  
Current balance:25000  
1.  
    pen new account  
2.  
    redit in your account  
3.  
    ithdraw from your account  
4.  
    how balance  
5.  
    xit  
Enter your choice  
4  
accno:2973  
Current balance:25000
```

**NOTES**

```
1.
    pen new account
2.
    redit in your account
3.
    ithdraw from your account
4.
    how balance
5.
    xit
Enter your choice
5
```

---

**3.3 Multilevel inheritance, Interfaces and Packages:**

---

**18. Program to implement inheritance and demonstrate use of method overriding.**

---

```
import java.lang.*;
class A
{
void display()
{
System.out.println("This is from class A ");
}}
class B extends A
{
void display()
{
System.out.println("This is from class B ");
}}
class AB
{
public static void main(String arg[]) {
```

```
B obj=new B();
obj.display();
} }
```

**Output:**

```
C:\jdk1.3\bin>javacAB.java
C:\jdk1.3\bin>javaAB
Thisisfromclass B
```

---

**19.Programtoimplementinheritancebyapplyingvariousaccess controlstoits datamembers andmethods.**


---

```
importjava.io.DataInputStream;
class Student
{
privateintrollno;
private String name;
DataInputStream dis=new DataInputStream(System.in);
public void getrollno()
{
try
{
System.out.println("Enter rollno");
rollno=Integer.parseInt(dis.readLine());
System.out.println("Enter name ");
name=dis.readLine();
}
catch(Exception e){ } }
voidputrollno()
{
System.out.println("Roll No =" +rollno);
```



**NOTES**

```

System.out.println("Name =" + name);
    }
}

class Marks extends Student {
protected int m1,m2,m3;
void getmarks()
{
try
{
System.out.println("Enter marks :");
m1=Integer.parseInt(dis.readLine());
m2=Integer.parseInt(dis.readLine());
m3=Integer.parseInt(dis.readLine());
}
catch(Exception e) { } }
void putmarks() {
System.out.println("m1=" + m1);
System.out.println("m2=" + m2);
System.out.println("m3=" + m3);
}
}

class Result extends Marks {
private float total;
void compute_display() {
total=m1+m2+m3;
System.out.println("Total marks : " + total);
} }
class MultilevelDemo
{

```

**NOTES**

```

public static void main(String args[])
{
Result r=new Result();
r.getrollno();
r.getmarks();
r.putrollno();
r.putmarks(); r.compute_display();
} }

```

**Output:**

```

C:\jdk1.3\bin>javacMultilevelDemo.java

C:\jdk1.3\bin>javaMultilevelDemo
Enter rollno12345
EnternameAvinash
Entermarks :
54
78
46
RollNo=12345
Name=Avinash
m1=54
m2=78
m3=46
Totalmarks:178.0

```

**20.Programtodemonstrateuseof implementinginterfaces.**

```

importjava.lang.*;
interface Area
{
final static float pi=3.14F;
float compute(float x,float y);
}
class rectangle implements Area

```

**NOTES**

```
{
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);
}
}
classinterfacedemo
{
public static void main(String a[])
{
rectanglerect=new rectangle();
circlecir=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:\jdk1.3\bin>javacinterfacedemo.java
```

```
C:\jdk1.3\bin>javainterfacedemo
```

```
Areaofrectangle=628.0
```

```
Areaofcircle=2827.43
```

---

**21.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 args[])
    {
        rectangle rect=new rectangle();
        double result=rect.compute(10.2,12.3);
        rect.display_result(result);
    } }
```

**NOTES**

## NOTES

**Output:**

```
C:\jdk1.3\bin>javac InterfaceExtendsDemo.java
C:\jdk1.3\bin>java InterfaceExtendsDemo
TheAreaais :393.9444131612778
```

---

**22. Program to implement the concept of importing classes from user defined package and creating packages.**


---

```
/*Source code of package p1 under the directory C:\jdk1.3\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:\jdk1.3\bin>edit StudentTest.java
*/

import p1.*;
class StudentTest {
public static void main(String args[])
{
```

**NOTES**

```

Student s=new Student();
s.getdata(123,"xyz");
s.putdata();
}
}

```

**Output:**

```

C:\jdk1.3\bin>javacp1\Student.java
C:\jdk1.3\bin>javac StudentTest.java
C:\jdk1.3\bin>javaStudentTest
regno=123
name =xyz

```

**23. User defined package:**

/\*Source code of package package1 under the directory

```
C:\jdk1.3\bin>package1\edit cal.java */
```

```

package package1;
public class cal
{
public double volume(double h,doublew,double d)
{
return(h*w*d);
}
publicint add(intx,int y)
{
return(x+y);}

publicintdivide(intx,int y)
{
return(x/y);
}
}

```

**NOTES**

```
/* Source code of the main function under C:\jdk1.3\bin>edit
```

```
packagedemo.java */
```

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

```
import package1.cal;
```

```
classpackagedemo
```

```
{
```

```
public static void main(String args[])
```

```
{
```

```
calcalc=new cal();
```

```
int sum=calc.add(10,20);
```

```
doublevol=calc.volume(10.4,13.26,32.326);
```

```
int div=calc.divide(20,4);
```

```
System.out.println("ADD"+sum);
```

```
System.out.println("VOLUME"+vol);
```

```
System.out.println("Division"+div);
```

```
}
```

```
}
```

**Output:**

```
C:\jdk1.3\bin>javac package\cal.java
```

```
C:\jdk1.3\bin>javac packagedemo.java
```

```
C:\jdk1.3\bin>java packagedemo
```

```
ADD 30
```

```
VOLUME 4457.884704
```

```
Division 5
```

**24. Message Passing Through methods.**

```
public class MsgPass_Methods
{
voiddisplayInt(int x, int y)
{
int z = x + y;System.out.println("Int Value is : "+z);
}
voiddisplayFloat(float x, float y)
{

float z = x * y;
System.out.println("Float Value is : "+z);
}
}
classMsgPassExample {
public static void main (String args[])
{
MsgPass_Methodsobj = new MsgPass_Methods();
obj.displayInt(10, 20);
obj.displayFloat((float)2.35, (float)5.89);
}
}
```

**Output**

Int Value is : 30

Float Value is : 13.841499

**NOTES**



## NOTES

---

## CHAPTER IV

### BLOCK III: THREAD

#### 4.1 THREAD:

#### 25. 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 started:");
        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 started:");
        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 started:");
        for(int k=1;k<=5;k++)
        {
```

**NOTES**

```
System.out.println("\t from thread C:k="+k); }  
System.out.println("exit from thread C:"); }  
}  
class Threadtest {  
public static void main(String args[]) {  
new A().start();  
new B().start();  
new C().start();  
} }  
}
```

**Output:**

```
C:\jdk1.3\bin>javac Threadtest.java  
C:\jdk1.3\bin>java Threadtest  
thread A is started:  
thread B is started:  
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:
```

## NOTES

---

**26. 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 args[]) {
        X R = new X();
        Thread T = new Thread(R);
        T.start();
    }
}
```

**Output:**

```
C:\jdk1.3\bin>javac Runnabletest.java
C:\jdk1.3\bin>java Runnabletest
ThreadX:1
ThreadX:2
ThreadX:3
ThreadX:4
ThreadX:5
ThreadX:6
ThreadX:7
ThreadX:8
ThreadX:9
End of ThreadX
```

**Note: Java does not support virtual function – It can be implemented using C++**

---

## CHAPTER V

---

# BLOCK IV: I/O AND EXCEPTION HANDLING

---

---

### 5.1. Input / Output Operations:

#### 27. Program to demonstrate I/O operations

---

```
import java.io.*;

class file
{
public static void main(String args[]) throws IOException
{
intempno,salary;
String name, design;
int more;
FileOutputStreamfos=new FileOutputStream("emp.java");
PrintWriter write=new PrintWriter(fos);
DataInputStream ds=new DataInputStream(System.in);
do
{
System.out.println("Enter employee no:");
empno=Integer.parseInt(ds.readLine());
System.out.println("Enter employee name:");
name=(ds.readLine());
System.out.println("Enter employee salary no:");
salary=Integer.parseInt(ds.readLine());
System.out.println("Enter designation:");
design=(ds.readLine());
write.println(empno+"\t" +name+ "\t" +design+"\t"+salary);
System.out.println("add more records=1,exit=0");
```

**NOTES**

```

more=Integer.parseInt(ds.readLine());
}
while(more==1);
write.close();
}
}

```

**Output:**

```

Enter employee no:
101
Enter employee name:
Kalai
Enter employee salary
20000
Enter designation
Manager
Add more records=1, exit =0
1
Enter employee no:
102
Enter employee name:
raja
Enter employee salary
25000
Enter designation
Accounts officer
Add more records=1, exit =0
0

```

---

**28. Program to demonstrate file read operation:**

---

```

import java.io.*;
class fileread
{
public static void main(String args[])throws IOException
{

```

**NOTES**

```

intempno,salary;
String name,design;
FileInputStreamfis=new FileInputStream("emp.java");
InputStreamReaderisr=new InputStreamReader(fis);
StreamTokenizer tokens=new StreamTokenizer(isr);
while(tokens.nextToken()!=tokens.TT_EOF)
{
empno=(int)tokens.nval;
tokens.nextToken();
name=tokens.sval;
tokens.nextToken();
salary=(int)tokens.nval;
design=tokens.sval;
tokens.nextToken();
salary=(int)tokens.nval;
System.out.println(empno+" "+name+" "+salary+" "+design);
}
}
}

```

**Output:**

```

C:\jdk1.3\bin>javac fileread.java

C:\jdk1.3\bin>java fileread
101 Kalai 20000 Manager
102 Raja 25000 Accounts officer

```

**5.2 Exception Handling****29. Program to implement the concept of Exception Handling using predefined exception.**

```

importjava.lang.*;
classException_handle
{
public static void main(String args[]) {
int a=10,b=5,c=5,x,y;

```

**NOTES**

```

try
{
x=a/(b-c);
}
catch(ArithmeticException e)
{
System.out.println("DIVISION BY ZERO");
}
y=a/(b+c);
System.out.println("y="+y);
}
}

```

**Output:**

```

C:\jdk1.3\bin>javacException_handle.java
C:\jdk1.3\bin>javaException_handle

DIVISION BY ZERO y=1

```

---

**30. Program to implement the concept of Exception Handling.**


---

```

import java.io.*;
import java.lang.*;
import java.util.*;
class except
{
public static void main(String args[])throws IOException
{
intch;
DataInputStream ds=new DataInputStream(System.in);
do
{

```

**NOTES**

```
System.out.println("menu");
System.out.println("1.add");
System.out.println("2.str");
System.out.println("3.array");
System.out.println("4.exit");
System.out.println("Enter your choice");
ch=Integer.parseInt(ds.readLine());
switch(ch)
{
case 1:
int a=5,b=0,c;
try
{
c=a/b;
System.out.println("Result"+c);
}
catch(Exception e)
{

System.out.println(e.toString());
e.printStackTrace();
}
break;
case 2:
String s="abcds";
try
{
System.out.println(s.charAt(5));
}
catch(Exception e)
{
System.out.println(e.toString());
```



**NOTES**

```

e.printStackTrace();
}
break;
case 3:
int x[]=new int[5];
try
{
System.out.println(x[5]);
}
catch(Exception e)
{
System.out.println(e.toString());
e.printStackTrace();
}
break;
case 4:
break;

}}
while(ch<5);
}
}

```

**Output:**

```

C:\jdk1.3\bin>javac except.java
C:\jdk1.3\bin>java except
Menu
1.add
2.str
3.array
4.exit
Enter your choice
1
java.lang.ArithmeticException: / by zero
java.lang.ArithmeticException: / by zero

```

**NOTES**

```

atexcept.main(except.java:25)
Menu
1.add
2.str
3.array
4.exit
Enter your choice
2
java.lang.StringIndexOutOfBoundsException: String index out of range: 5
java.lang.StringIndexOutOfBoundsException: String index out of range: 5
atjava.lang.String.charAt(String.java:658)
atexcept.main(except.java:38)
Menu
1.add
2.str
3.array
4.exit
Enter your choice
3
java.lang.ArrayIndexOutOfBoundsException
java.lang.ArrayIndexOutOfBoundsException
atexcept.main(Compiled Code)
Menu
1.add
2.str
3.array
4.exit
Enter your choice
4

```

**31. 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 {

```

**NOTES**

```
MyException(String message)
{
super(message); }
}
classuserdef {
public static void main(String args[])
{
int age;
DataInputStream ds=new DataInputStream(System.in);
try
{
System.out.println("Enter the age (above 15 and 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:**

```
c:\jdk1.3\bin>java userdef

Enter the age (above 15 and below 25) : 6
Caught MyException Number not in range

c:\jdk1.3\bin>java userdef

Enter the age (above 15 and below 25) : 20
the number is :20
```

**NOTES**

## NOTES

---

## CHAPTER VI

### BLOCK V: NETWORK PROGRAMMING

#### 6.1 Applet Programs:

---

##### 32. Program to demonstrate Keyboard event

---

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

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

public class KeyEvents extends Applet implements KeyListener {
    String msg = " ";
    int x=10,y=20;
    public void init()
    {
        addKeyListener(this);
        requestFocus();
    }

    public void keyPressed(KeyEvent k)
    {
        showStatus("key down"); }

    public void keyReleased(KeyEvent k)
    {
        showStatus("key up"); }

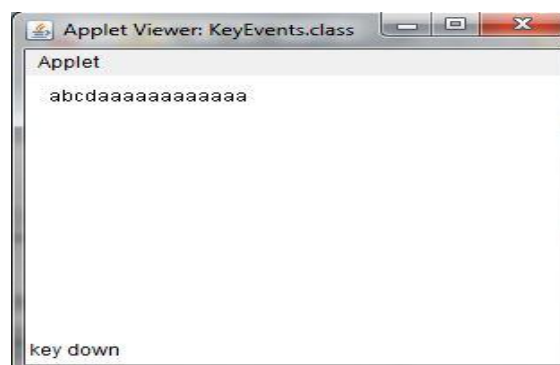
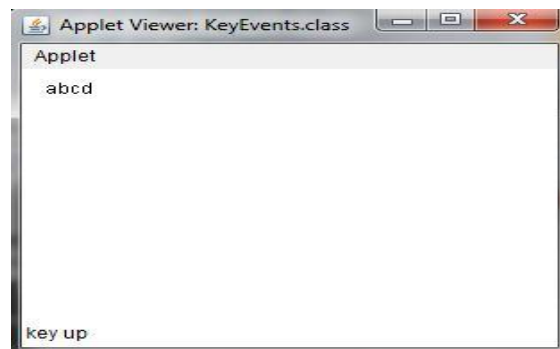
    public void keyTyped(KeyEvent k)
    {
```

**NOTES**

```
msg +=k.getKeyChar();  
repaint();  
}  
public void paint(Graphics g)  
{  
g.drawString(msg,x,y); }  
}
```

**Output**

```
C:\jdk1.3\bin>javacKeyevents.java  
C:\jdk1.3\bin>appletviewerKeyevents.java
```



## NOTES

---

**33. 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();
    }
}
```

**NOTES**

```
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 ="*";
setStatus("Dragged mouse at " +x+ " & "+y);
repaint();
}

public void mouseMoved(MouseEvent m) {
setStatus("Moving mouse at " +m.getX()+ " & "+m.getY()); }

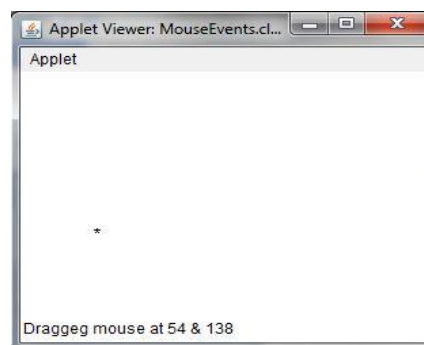
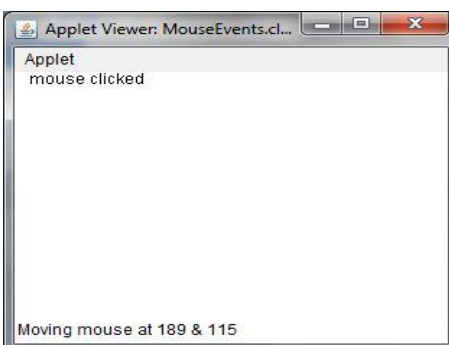
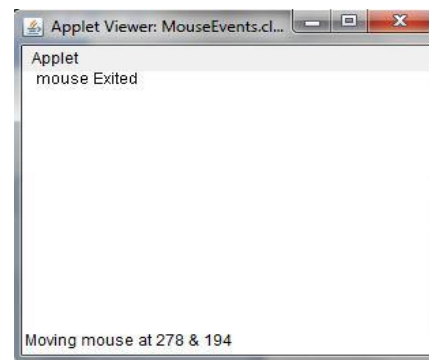
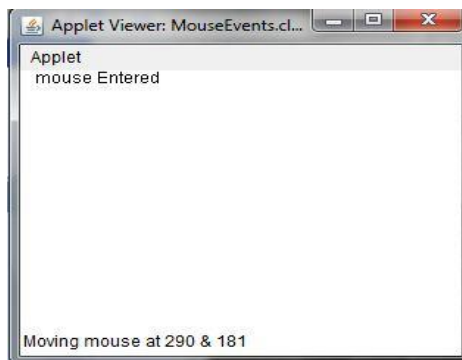
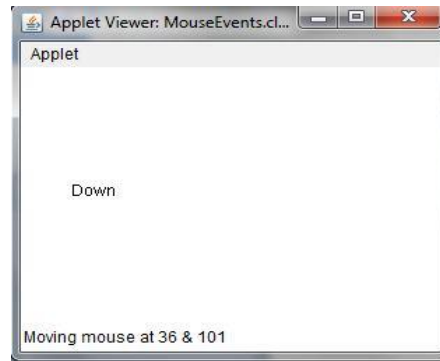
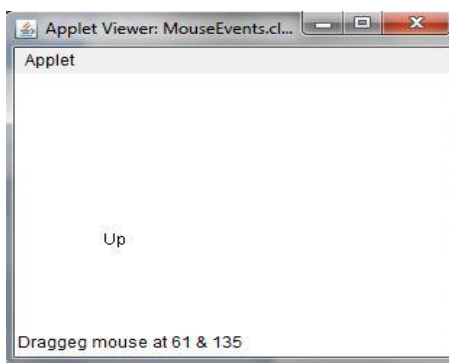
public void paint(Graphics g) {
g.drawString(msg,x,y); }
}
```



NOTES

**Output:**

```
C:\jdk1.3\bin>javacMouseEvents.java  
C:\jdk1.3\bin>appletviewerMouseEvents.java
```



---

**34. Write program for using Graphics class**

---

**NOTES**

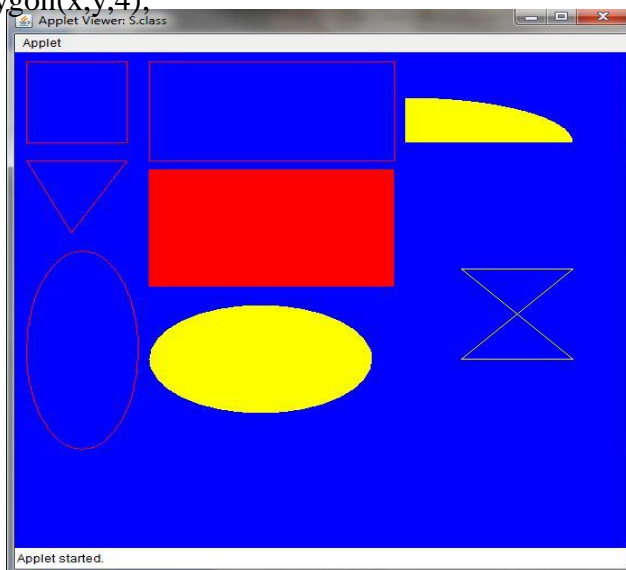
```
import java.applet.*;
import java.awt.*;

/* <applet code="Shapes.class" width=800 height=800></applet>*/

public class Shapes extends Applet {
    public void paint(Graphics g) {
        setForeground(Color.red);
        setBackground(Color.blue);
        //drawing squares
        g.drawLine(10,10,100,10);
        g.drawLine(10,10,10,10);
        g.drawLine(10,100,100,100);
        g.drawLine(100,100,100,10);
        // Drawing triangle
        g.drawLine(10,120,100,120);
        g.drawLine(10,120,50,200);
        g.drawLine(50,200,100,120);
        //drawing Rectangle
        g.drawRect(120,10,220,120);
        g.fillRect(120,120,220,120);
        //drawing ellipse and circle
        g.drawOval(10,220,100,220);
        g.setColor(Color.yellow);
        g.fillOval(120,250,250,250);
        //draw a filled arc
        g.fillArc(350,50,400,100,0,90);
        //draw a polygon
        int x[]={400,500,400,500};
```

## NOTES

```
int y[]={240,240,340,340};
g.drawPolygon(x,y,4);
} }
```

**Output:**


---

### 35. Applet Program for displaying circles

---

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

/* <applet code="circle2.class" width=800 height=800></applet>*/

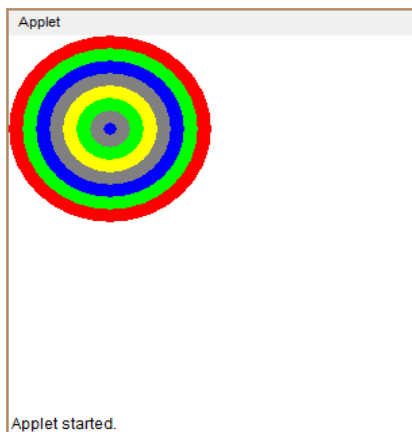
public class circle2 extends Applet
{
    public void paint(Graphics g)
    {
        Color colors[]={Color.red,Color.green,Color.blue,Color.gray,
            Color.yellow,Color.green,Color.gray,Color.blue,Color.red,Color.pink};
        int x=(getSize().width)/2;
        int y=(getSize().height)/2;
        for(int i=0;i<5;i++)
```

**NOTES**

```

{
for(i=0;i<=colors.length;i++)
{
int j=i*10;
int k=i*20;
g.setColor(colors[i]);
g.fillOval(x+j-150,y+j-150,150-k,150-k);
}
}
g.drawString("width"+getSize().width,10,10);
g.drawString("height"+getSize().height,70,10);
}
}

```

**Output:****6.2 Network Programs****36. Program to display IP address of the given website**

```

import java.net.*;
import java.io.*;
public class ip
{
public static void main ( String[] args ) throws IOException
{

```

**NOTES**

```
String hostname = args[0];
try
{
    InetAddress ipaddress = InetAddress.getByName(hostname);
    System.out.println("IP address: " + ipaddress.getHostAddress());
}
catch ( UnknownHostException e )
{
    System.out.println("Could not find IP address for: " + hostname);
}
}}
```

**Output:**

```
C:\jdk1.3\bin>javacip.java
C:\jdk1.3\bin> java ipwww.google.com

IP address: 136.206.217.25
```

---

**37. Program to display IP address of the localhost**

---

```
import java.net.InetAddress;
import java.net.UnknownHostException;

public class Demo {
    public static void main(String[] args) {
        InetAddress ipadd;
        String hostname;
        try {
            ipadd = InetAddress.getLocalHost();
            hostname = ipadd.getHostName();
            System.out.println("Your IP address : " + ipadd);
            System.out.println("Your Hostname : " + hostname);
        }
    }
}
```

```
} catch (UnknownHostException e) {  
    }  
}
```

**Output:**

```
C:\jdk1.3\bin>javacDemo.java  
  
C:\jdk1.3\bin> java Demo  
  
Your IP address : 4d623edc62d4/172.17.0.2  
  
Your Hostname : 4d623edc62d4
```

---

**38. Program to display parts of URL address**

---

```
import java.net.URL;  
public class Main  
{  
    public static void main(String[] args) throws Exception  
    {  
        URL url = new URL(args[0]);  
        System.out.println("URL is " + url.toString());  
        System.out.println("protocol is " + url.getProtocol());  
        System.out.println("file name is " + url.getFile());  
        System.out.println("host is " + url.getHost());  
        System.out.println("path is " + url.getPath());  
        System.out.println("port is " + url.getPort());  
        System.out.println("default port is " + url.getDefaultPort());  
    }  
}
```

**NOTES**

**Output:**

```
C:\jdk1.3\bin>javac Main.java
C:\jdk1.3\bin>javaMain https://www.alagappauniversity.ac.in/page/219

URL is https://www.alagappauniversity.ac.in/page/219
protocol is https:
file name is /page/219
host is www.alagappauniversity.ac.in
path is /page/219
port is -1
default port is 443
```

**Model Question Paper**  
**DISTANCE EDUCATION**  
**B.Sc. (INFORMATION TECHNOLOGY) DEGREE EXAMINATION**  
**CBCS – (2018-2019 Academic Year onwards)**  
**Third Semester**

**SubCode12934 - Internet and Java Programming Lab**

**Time: Three hours**

**Maximum: 75 marks**

**One question should be given to each candidate by Lot system**

1. (a) Write a Java Program to define a class, describe its constructor, overload the Constructors and instantiate its object  
(b) Write a Java program for using Graphics class  
-to display basic shapes and fill them  
-draw different items using basic shapes  
-set background and foreground colors
2. (a) Write a Java Program to display the Fibonacci Series  
(b) Write a Java Program to demonstrate Mouse events using applet
3. (a) Write a Java Program to display alphabetical order for the given set of names  
(b) Write a program to implement the concept of Exception Handling by creating user-defined exceptions.
4. (a) Write a Java Program to check whether the given number is Prime or not  
(b) Write a program to implement the concept of threading by implementing Runnable Interface
5. (a) Write a Java Program to display ascending and descending order of the given set of numbers.  
(b) Write a Java program to implement the concept of importing classes from user-defined package and creating packages.
6. (a) Write a Java program to practice using String class and its methods.  
(b) Write a Java Program to implement multi-level inheritance by applying various access controls to its data members and methods.
7. (a) Write a Java program to implement the concept of Exception Handling.  
(b) Write a program to demonstrate use of extending interfaces.
8. (a) Write a Java Program to define a class, define instance methods and overload them and use them for dynamic method invocation  
(b) Write a program to implement the concept of threading by extending Thread Class

\*\*\*\*\*

*Self-Instructional Material*