॥ अप्पो दीपो भव ॥
महात्मा गांधी मिशन विश्वविद्यालय
MGM UNIVERSITY

# Jawaharlal Nehru Engineering College

Accredited with ‘A’ grade by NAAC, ISO 9001:2015
certified, AICTE Approved

# Laboratory Manual

## Object Oriented Programming Using Java

For

FY MCA Part-1

## Department of MCA

# LABORATORY MANUAL CONTENTS

This manual is intended for the First year students of MCA in the subject of Object Oriented Programming using JAVA.

This manual typically contains practical/Lab Sessions related JAVA Programming covering various aspects related the subject to enhanced understanding.

Students are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions

| **Prepared by** | **Reviewed by** | **Approved by** |
|---|---|---|
| A. A. Dudhgaonkar | Dr. S. S. Deshmukh | Dr. H. H. Shinde |
| Assistant Professor | Head of Department | Principal |

# FORWARD

It is my great pleasure to present this laboratory manual for First year students for the subject of Java Programming.
As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.

As you may be aware that MGM has already been awarded with ISO 9001-2015 certification and it is our endure to technically equip our students taking the advantage of the procedural aspects of ISO 9001-2015 Certification.

Faculty members are also advised that covering these aspects in initial stage itself, will greatly relieved them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear.

Dr. H.H. Shinde

Principal

## Institute Vision

To create self-reliant, continuous learner & competent technocrats imbued with human values.

==========================================================================

## Institute Mission

Imparting quality technical education to the students through participative teaching –learning process.
Developing competence amongst the students through academic learning and practical experimentation.
Inculcating social mindset and human values amongst the students.

==========================================================================

## Department Vision

Build a strong technical teaching and learning environment that responds swiftly to the challenges and needs of the current industry trends.

==========================================================================

## Department Mission

Provide excellent post graduate education in a state-of-the-art environment, preparing students for careers as computer technologist in self-employment, industry, government and of IT enabled sectors.

Support society by participating in and encouraging technology transfer.

==========================================================================

| PROGRAM OUTCOMES | |
|---|---|
| **PO No.** | **Program Outcome Description** |
| PO 1 | Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. |
| PO 2 | Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| PO 3 | Design / Development of solution: 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. |
| PO 4 | Conduct investigation 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. |
| PO 5 | 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. |
| PO 6 | The engineer & 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. |
| PO 7 | Environment & sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| PO 8 | Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| PO 9 | Individual & team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| PO 10 | 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, and give and receive clear instructions. |
| PO 11 | Project management & 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. |
| PO 12 | 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 Outcomes** | |
|---|---|
| PSO 1 | Apply knowledge and probability and statistics, including applications appropriate to computer engineering |
| PSO 2 | Inculcate and understand professional Ethics, Cyber, Social responsibility |
| PSO 3 | Assimilate knowledge of Mathematics and their applications in Cryptography, Graph Theory and Computing |
| PSO 4 | Understand Database, Data Mining, Normalization and Design Data Sensitive applications |
| PSO 5 | Imbibe knowledge to perform research on contemporary technology issues and publish research papers |
| PSO 6 | Perform feasibility study using OOSDLC for the problem domain |

| **Lab Outcomes** | |
|---|---|
| LO1 | Write object oriented programs |
| LO2 | Demonstrate Collection program. |
| LO3 | Demonstrate Multithreading program |
| LO4 | Write an applications that have an event-driven graphical user interface using the standard Java libraries &access database with JDBC . |

# SUBJECT INDEX

## DOs and DON'Ts in Laboratory:

1. Make entry in the Log Book as soon as you enter the Laboratory.

2. All the students should sit according to their roll numbers starting from their left to right.

3. All the students are supposed to enter the terminal number in the log book.

4. Do not change the terminal on which you are working.

5. All the students are expected to get at least the algorithm of the program/concept to be implemented.

6. Strictly observe the instructions given by the teacher/Lab Instructor.

## Instruction for Laboratory Teachers::

1. Submission related to whatever lab work has been completed should be done during the next lab session. The immediate arrangements for printouts related to submission on the day of practical assignments.

2. Students should be taught for taking the printouts under the observation of lab teacher.

3. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

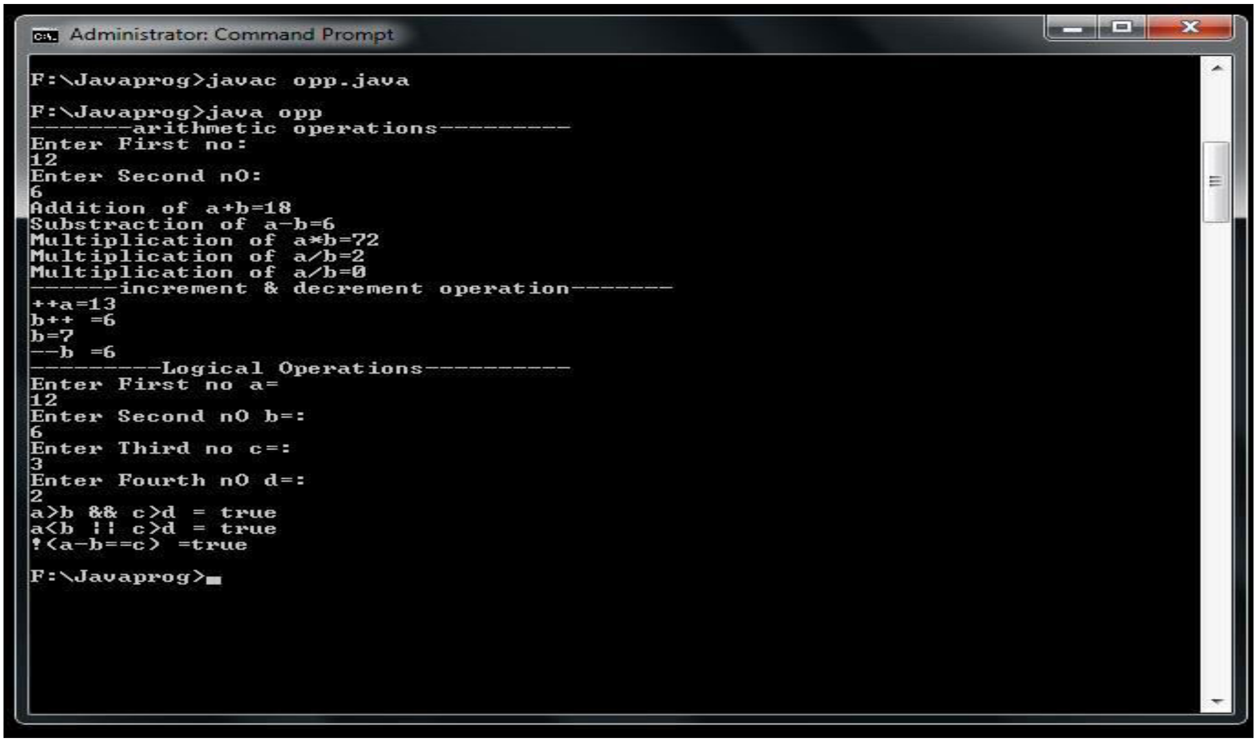
## Lab Exercise 1 (2 Hours)

**Aim**: Write a Program to demonstrate basic data type and operators in JAVA

**Algorithm:**

□ Declare and define a class. Define main method.
□ Declare variables and initialize variables.
□ Implement different operators in Java for e.g. Arithmetic, logical, conditional etc. with the help of variables.
□ Print all the values of variables, by using basic syntax of Java language.

**Output**:



```
F:\Javaprog>javac opp.java

F:\Javaprog>java opp
--------arithmetic operations----------
Enter First no:
12
Enter Second nO:
6
Addition of a+b=18
Substraction of a-b=6
Multiplication of a*b=72
Multiplication of a/b=2
Multiplication of a/b=0
------increment & decrement operation-------
++a=13
b++ =6
b=7
--b =6
---------Logical Operations----------
Enter First no a=
12
Enter Second nO b=:
6
Enter Third no c=:
3
Enter Fourth nO d=:
2
a>b && c>d = true
a<b || c>d = true
!(a-b==c) =true

F:\Javaprog>
```

**Exercises**:
1) Write a java program that prints all real solutions to the quadratic equation ax2+bx+c=0. Read in a, b, c and use the quadratic formula.
2) Write a program for addition of two using command line arguments.

## Lab Exercise 2 (2 Hours)

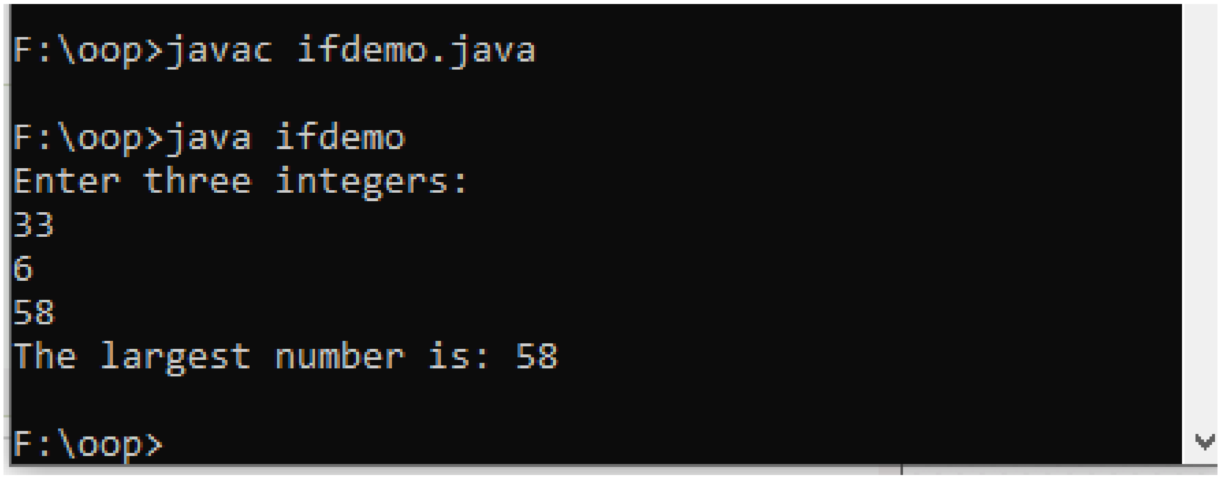
**Aim**: Program to demonstrate Decision Making Stmts in JAVA

**Problem Stmt**: WAP to find largest of three numbers

**Source code:**

```
import java.util.Scanner;
public class ifdemo
{
public static void main(String args[])
{
int num1, num2, num3;
System.out.println("Enter three integers: ");
Scanner in = new Scanner(System.in);
num1=in.nextInt();
num2=in.nextInt();
num3=in.nextInt();
if (num1 > num2 && num1 > num3)
System.out.println("The largest number is: "+num1);
else if (num2 > num1 && num2 > num3)
System.out.println("The largest number is: "+num2);
else if (num3 > num1 && num3 > num2)
System.out.println("The largest number is: "+num3);
else
System.out.println("The numbers are same.");
}
}
```

**Output:**



```
F:\oop>javac ifdemo.java

F:\oop>java ifdemo
Enter three integers:
33
6
58
The largest number is: 58

F:\oop>
```

**Exercise:**

WAP to check odd and even number
WAP to check prime number
WAP for Arithmetic calculator using switch case menu driven program
WAP Write a program to calculate the roots of Quadratic equations.

## Lab Exercise 3 (2 Hours)

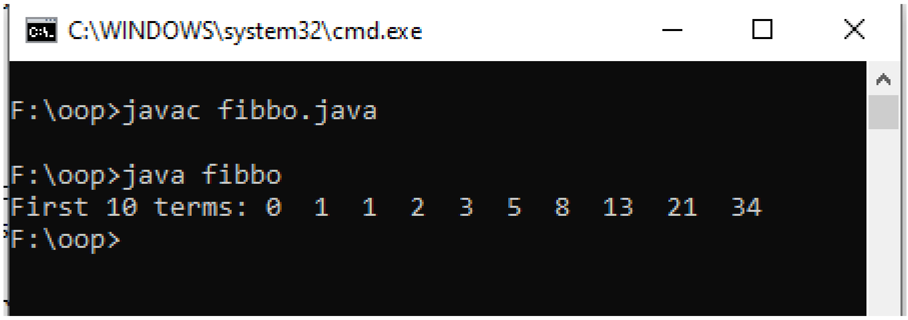
**Aim**: Program to demonstrate Loop Control Structures in JAVA

**Problem Stmt**: WAP to Fibonacci series

**Source code:**

```
public class fibbo
{

    public static void main(String[]  args) {

        int n = 10, t1 = 0, t2 = 1;
        System.out.print("First " + n + " terms: ");

        for (int i = 1; i <= n; ++i)
        {
            System.out.print(t1 + "  ");

            int sum = t1 + t2;
            t1 = t2;
            t2 = sum;
        }
    }
}
```

**Output:**



```
F:\oop>javac fibbo.java

F:\oop>java fibbo
First 10 terms: 0  1  1  2  3  5  8  13  21  34
F:\oop>
```

**Exercise:**

WAP for sum of digit
Write a program to print the following triangle

```
1      a       4321

12     ab       321

123    abc       21

1234   abcd       1
```

WAP for reverse number
WAP for palindrome number

## Lab Exercise 4 (2 Hours)

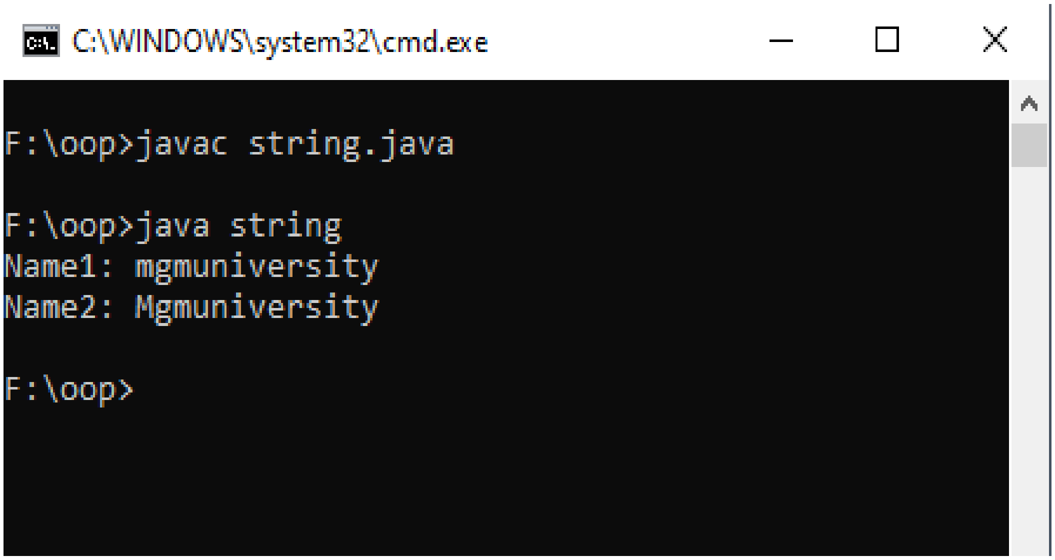
**Aim**: Program to demonstrate String class and its methods

**Problem Stmt:** Write a program to make the first letter of a String capital

**Source Code:**

```
class string{
  public static void main(String[] args) {

    // create a string
    String name = "mgmuniversity";
String n1=name;
    // create two substrings from name
    // first substring contains first letter of name
    // second substring contains remaining letters
    String firstLetter = name.substring(0, 1);
    String remainingLetters = name.substring(1, name.length());
    // change the first letter to uppercase
    firstLetter = firstLetter.toUpperCase();

    // join the two substrings
    name = firstLetter + remainingLetters;
    System.out.println("Name1:  " + n1);
    System.out.println("Name2:  " + name);

  }
}
```

**Output:**



```
F:\oop>javac string.java

F:\oop>java string
Name1: mgmuniversity
Name2: Mgmuniversity

F:\oop>
```

**Exercise:**

WAP Program to determine whether a given string is palindrome

WAP to count the total number of vowels and consonants in a string
WAP to perform string operations

## Lab Exercise 5 (2 Hours)

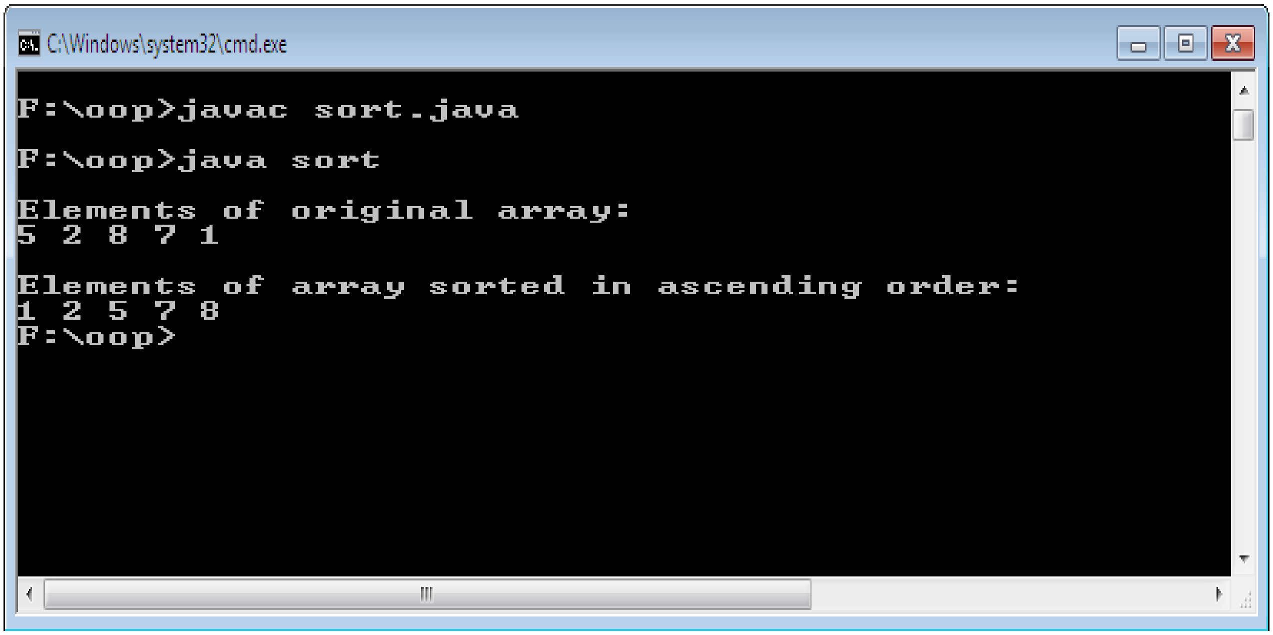
**Aim**: Program to demonstrate Array and is types

**Problem Stmt:** Write a Java Program to sort the elements of an array in ascending order

**Source Code:**

```
public class sort {
   public static void main(String[] args) {
              //Initialize array
      int [] arr = new int [] {5, 2, 8, 7, 1};
      int temp = 0;
      //Displaying elements of original array
      System.out.println("\nElements of original array: ");
      for (int i = 0; i < arr.length; i++) {
         System.out.print(arr[i] + " ");
      }
               //Sort the array in ascending order
      for (int i = 0; i < arr.length; i++) {
         for (int j = i+1; j < arr.length; j++) {
           if(arr[i] > arr[j]) {
              temp = arr[i];
              arr[i] = arr[j];
              arr[j] = temp;
           }
         }
      }
             System.out.println();
         //Displaying elements of array after sorting
      System.out.println("\nElements of array sorted in ascending order: ");

        for (int i = 0; i < arr.length; i++) {
         System.out.print(arr[i] + " ");
      }
   }
}
```

## Output:



```
F:\oop>javac sort.java

F:\oop>java sort

Elements of original array:
5 2 8 7 1

Elements of array sorted in ascending order:
1 2 5 7 8
F:\oop>
```

## Exercise:

WAP for calculating Matrix Multiplication operation
Write a Java Program for sorting a given list of names in ascending order.

WAP for linear search in JAVA

## Lab Exercise 6 (2 Hours)

**Aim**: Program to demonstrate use of class and objects

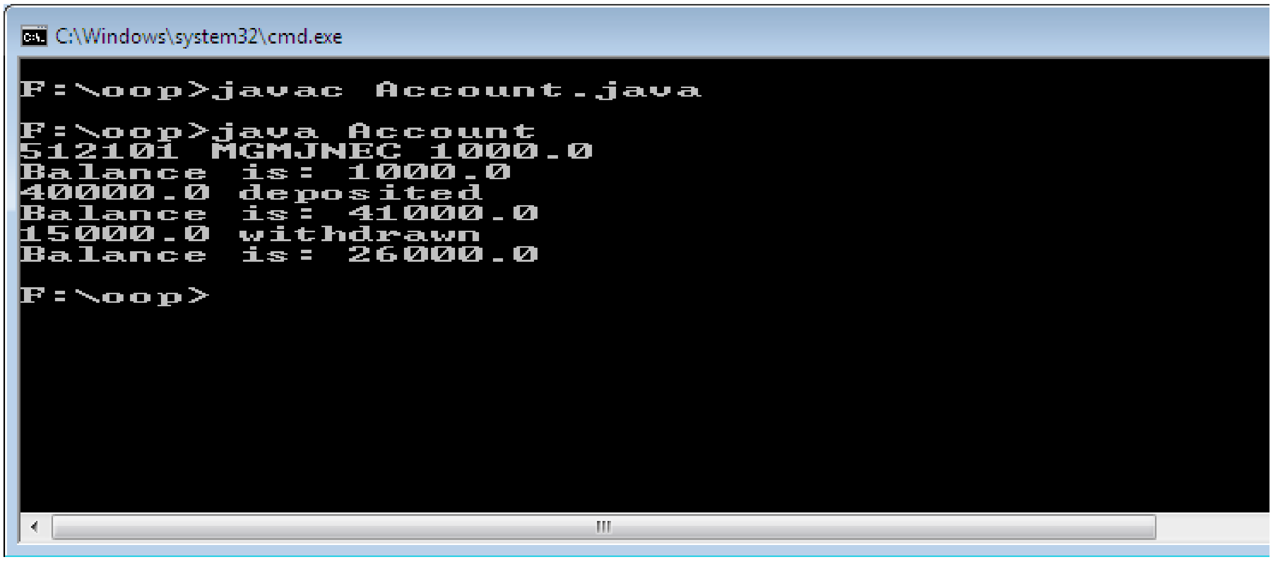
**Problem Stmt:** Write a Java Program to demonstrate the working of a banking- system where we deposit and withdraw amount from our account.

**Source Code:**

```
class bank

{

int acc_no;

String name;

float amount;

//Method to initialize  object

void insert(int a,String n,float amt)

{

acc_no=a;

name=n;

amount=amt;

}

//deposit method

void deposit(float amt)

{

amount=amount+amt;

System.out.println(amt+"  deposited");

}
```

```
//withdraw method
void withdraw(float amt)
{
if(amount<amt){
System.out.println("Insufficient Balance");
}
else
{
amount=amount-amt;
System.out.println(amt+" withdrawn");
}
}
//method to check the balance of the account
void checkBalance(){System.out.println("Balance is: "+amount);}
//method to display the values of an object
void display(){System.out.println(acc_no+" "+name+" "+amount);}
}

//Creating a test class to deposit and withdraw amount
class Account
{
public static void main(String[] args)
{ bank a1=new bank();
a1.insert(512101,"MGMJNEC",1000);
a1.display();
a1.checkBalance();
a1.deposit(40000);
a1.checkBalance();
a1.withdraw(15000);
a1.checkBalance();
}
}
```

**Output:**



```
F:\oop>javac Account.java

F:\oop>java Account
512101 MGMJNEC 1000.0
Balance is: 1000.0
400000.0 deposited
Balance is: 41000.0
150000.0 withdrawn
Balance is: 26000.0

F:\oop>
```

**Exercise:**

WAP using class & object for calculating area of circle, area of rectangle, area of triangle using menu driven

Write a program to create a room class, the attributes of this class is roomno, roomtype, roomarea and ACmachine. In this class the member functions are setdata and displaydata.

## Lab Exercise 7 (2 Hours)

**Aim** : Program to demonstrate use of Packages

**Problem Stmt:** Design a class to create package and use in another class

**Source Code:**

```
//save by A.java
package pack;
public class A{
  public void msg(){System.out.println("Hello");}
}
//save by B.java
package mypack;
import pack.*;
 class B{
 public static void main(String args[]){
   A obj = new A();
// pack.A obj = new pack.A();//using fully qualified name
   obj.msg();
  }
}
```

**Output:**

```
Hello
```

**Exercise:**

WAP to demonstrate Sub packages

WAP to demonstrate Access modifier

## Lab Exercise 8 (2 Hours)

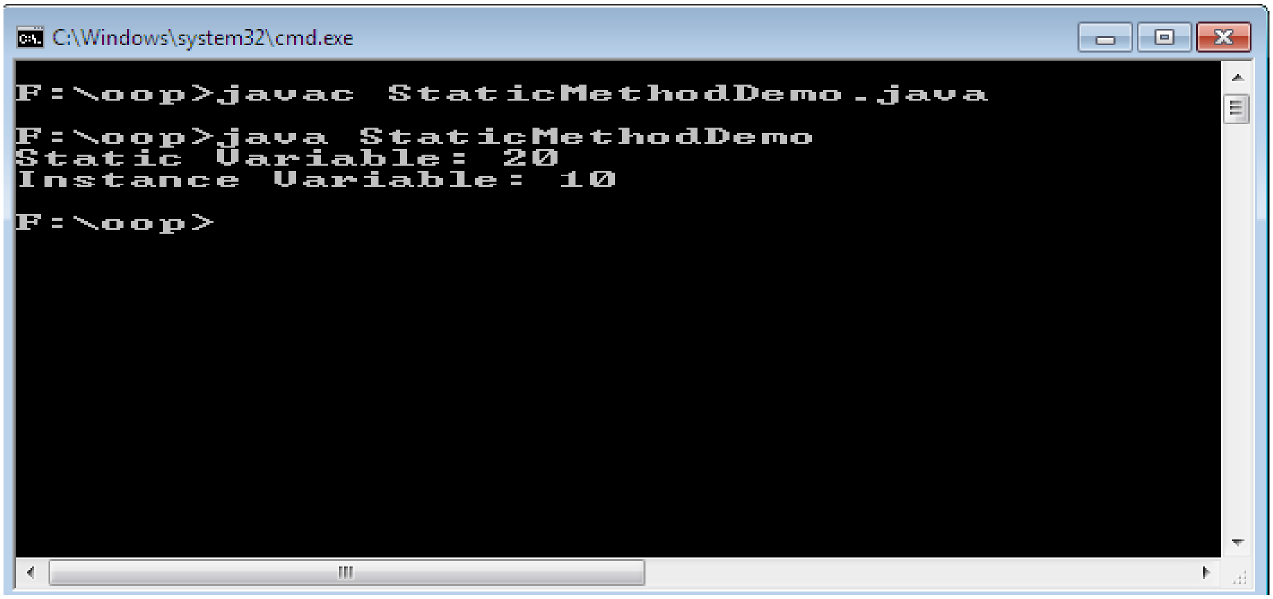
**Aim**: Program to demonstrate use of static variables, methods and block

**Problem Stmt:** WAP to print static and instance variable

**Source Code:**

```
class StaticMethod
{    int instVar = 10;
    static int staticVar = 20;
     public static void staticMethod ()
 {       //You can not access instVar directly.
      //  System.out.println(instVar);
//error: Cannot make a static reference to the non-static field instVar
         System.out.println("Static Variable: "+staticVar);
         StaticMethod smObject = new StaticMethod();
        System.out.println("Instance Variable: "+smObject.instVar);
//access instance variable
    }
}
 public class StaticMethodDemo
{
    public static void main(String[] args)
    {
        StaticMethod.staticMethod(); //call Static method
    } }
```

## Output



```
F:\oop>javac StaticMethodDemo.java

F:\oop>java StaticMethodDemo
Static Variable: 20
Instance Variable: 10

F:\oop>
```

## Exercise:

WAP Program to illustrate the use of static variable (Count ) which is shared with all objects.

WAP to get the cube of a given number using the static method

WAP to demonstrate use of Static Block

## Lab Exercise 9 (2 Hours)

**Aim**: Program to demonstrate constructor overloading

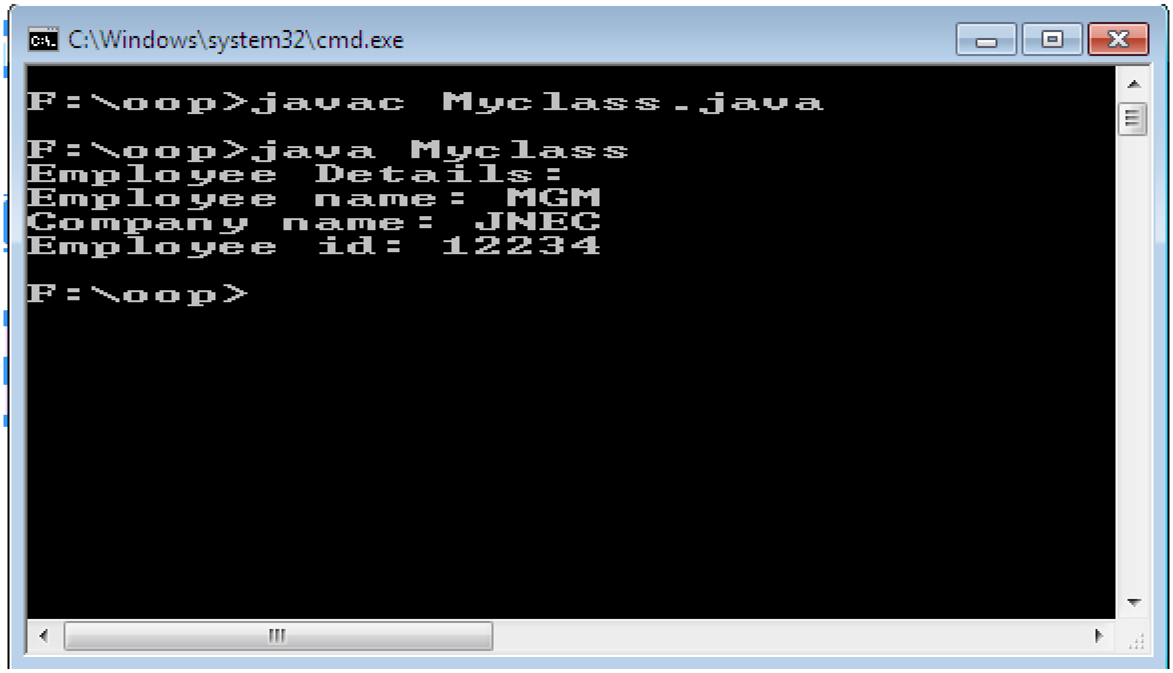
**Problem Stmt:** Write a program for employee class using constructors

**Source Code:**

```
class Employee

 {

        Employee()

        {

         System.out.println("Employee Details:");

        }

        Employee(String name)

        {

          System.out.println("Employee name: " +name);

        }

        Employee(String nCompany, int id)

        {

        System.out.println("Company name: " +nCompany);

        System.out.println("Employee id: " +id);

        }

 }
```

```
class Myclass
{   public static void main(String[]  args)
  {
     Employee emp = new Employee();
     Employee emp2 = new Employee("MGM");
     Employee emp3 = new Employee("JNEC", 12234);
  }
}
```

**Output:**



```
F:\oop>javac Myclass.java

F:\oop>java Myclass
Employee Details:
Employee name: MGM
Company name: JNEC
Employee id: 12234

F:\oop>
```

**Exercise:**

WAP to illustrated Constructor overloading using This keyword

WAP to illustrated final keyword in constructor

## Lab Exercise 10 (2 Hours)

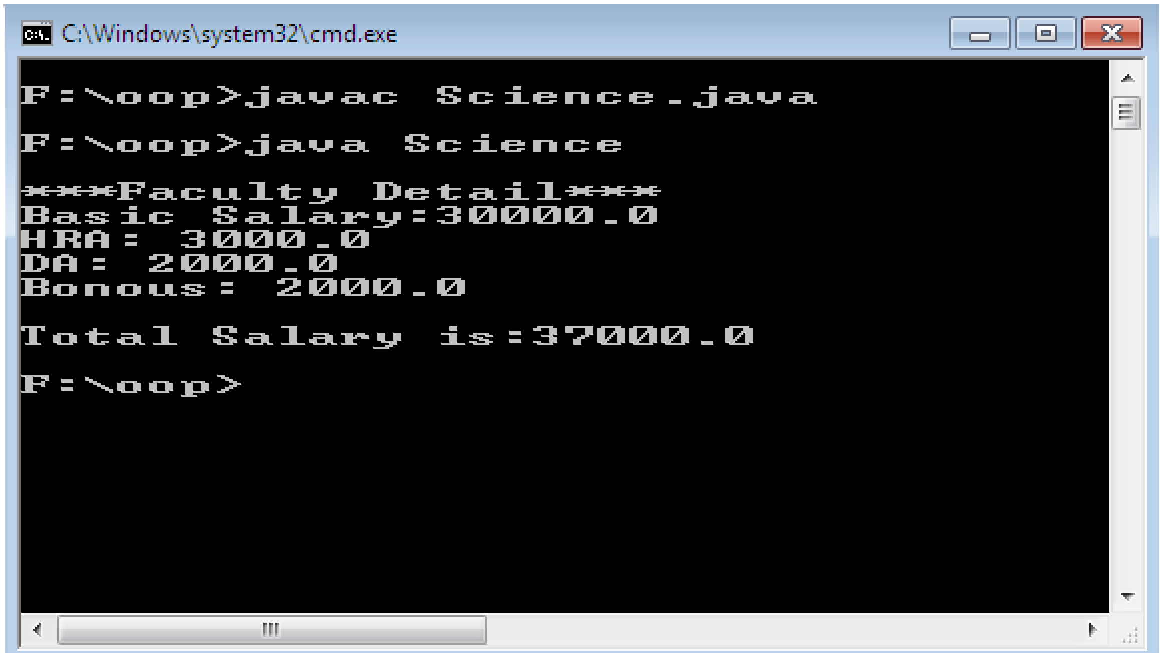
**Aim**: Program to demonstrate use of inheritance

**Problem Stmt:** WAP to calculate total salary of faculty of college including hra, da, bonous using multilevel inheritance

**Source Code:**

```
//multilevel  inheritance

class Faculty

{  float total_sal=0, salary=30000;  }

class HRA extends Faculty

{  float hra=3000;   }

class DA extends HRA

{  float da=2000;  }

class Science extends DA

{

float bonous=2000;

public static void main(String  args[])

{   Science obj=new Science();

obj.total_sal=obj.salary+obj.hra+obj.da+obj.bonous;

System.out.println("\n***Faculty  Detail***\nBasic Salary:"+obj.salary+"\nHRA:  "+obj.hra+"\nDA: "+obj.da+"\nBonous: "+obj.bonous);

System.out.println("\nTotal  Salary is:"+obj.total_sal);

}

}
```

## Output:



```
F:\oop>javac Science.java

F:\oop>java Science

***Faculty Detail***
Basic Salary:30000.0
HRA: 3000.0
DA: 2000.0
Bonous: 2000.0

Total Salary is:37000.0

F:\oop>
```

## Exercise:

WAP to illustrate use of hierarchical inheritance

WAP to illustrate use of Hybrid Inheritance

WAP to illustrate use of Super Keyword

## Lab Exercise 11 (2 Hours)

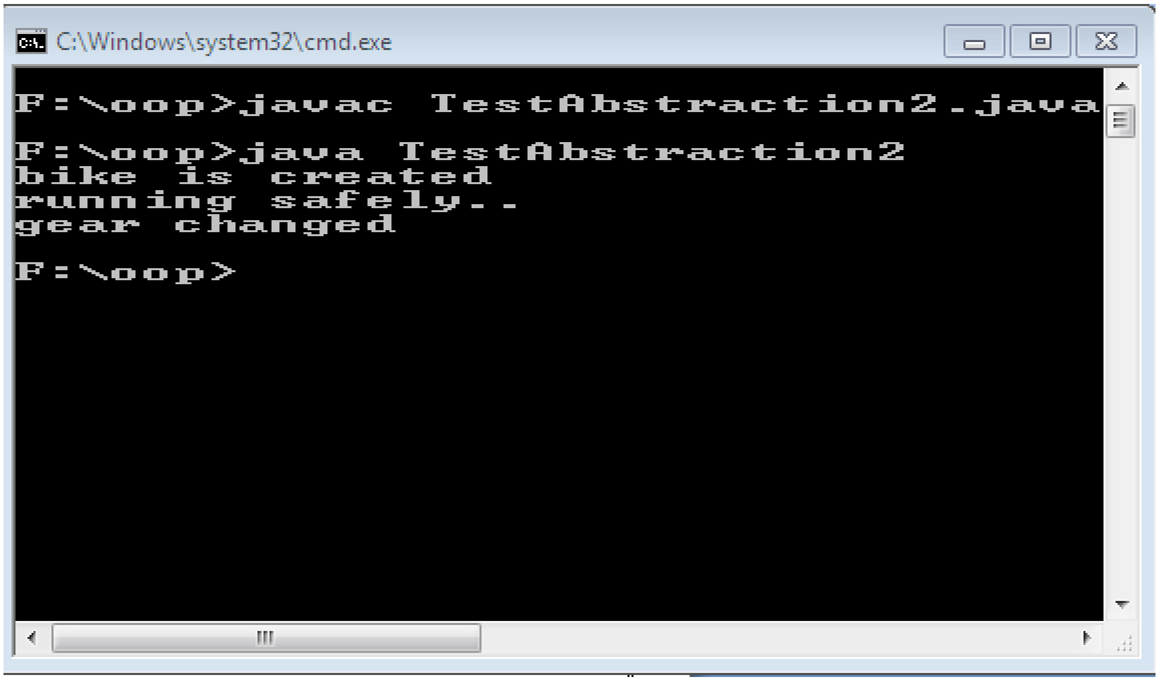
**Aim**: Program to demonstrate use of Abstract class and method

**Problem Stmt:** WAP for Bike an abstract class that has abstract and non-abstract methods

**Sorce Code**:

```
//Example of an abstract class that has abstract and non-abstract
methods

 abstract class Bike{

   Bike(){System.out.println("bike is created");}

   abstract void run();

   void changeGear()
{System.out.println("gear changed");}

 }

//Creating a Child class which inherits Abstract class

 class Honda extends Bike{

 void run(){System.out.println("running safely..");}

 }

//Creating a Test class which calls abstract and non-abstract methods

 class TestAbstraction2{

 public static void main(String args[])

{    Bike obj = new Honda();

  obj.run();

  obj.changeGear();

 }  }
```

**Output:**



```
F:\oop>javac TestAbstraction2.java

F:\oop>java TestAbstraction2
bike is created
running safely..
gear changed

F:\oop>
```

**Exercise:**

Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape

## Lab Exercise 12 (2 Hours)

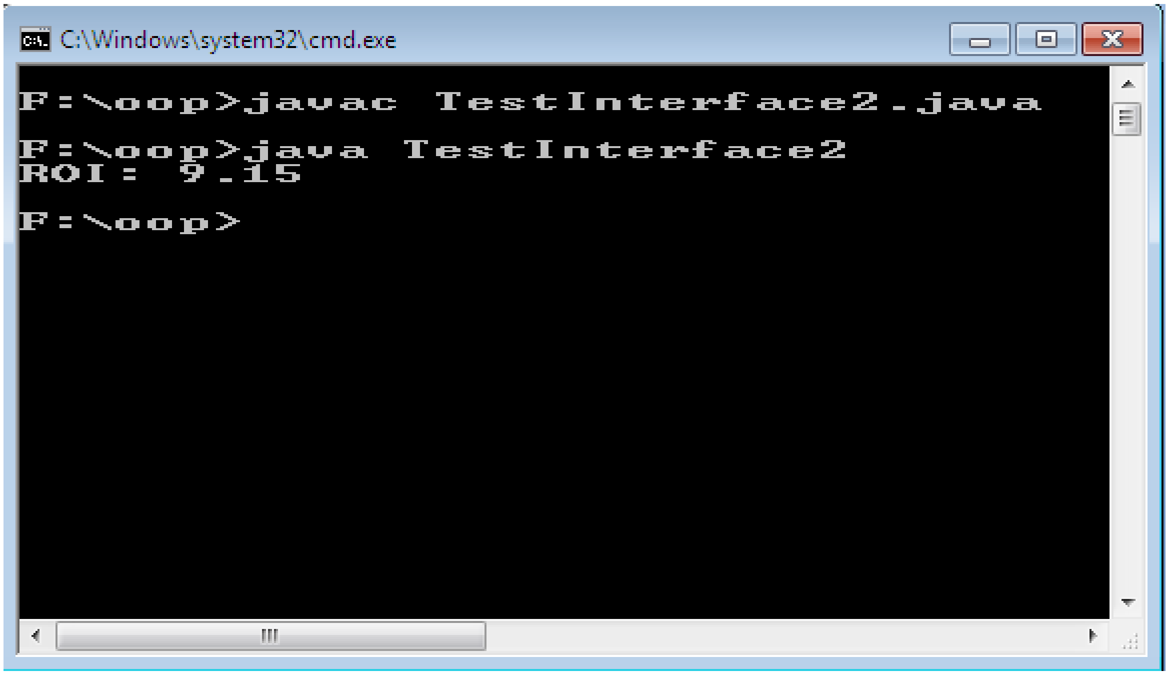
**Aim**: Program to demonstrate use of Interface

**Problem Stmt:** WAP to display BANK Rate of Interest using Interface

**Source Code:**

```
interface Bank{

float rateOfInterest();

}

class SBI implements Bank{

public float rateOfInterest()
{return 9.15f;}

}

class PNB implements Bank{

public float rateOfInterest()
{return 9.7f;}

}

class TestInterface2{

public static void main(String[] args){

Bank b=new SBI();

System.out.println("ROI: "+b.rateOfInterest());

}}
```

**Output:**



```
F:\oop>javac TestInterface2.java

F:\oop>java TestInterface2
ROI : 9.15

F:\oop>
```

**Exercise:** Write a program to create interface named test. In this interface the member function is square. Implement this interface in arithmetic class. Create one new class called ToTestInt in this class use the object of arithmetic class.

Create an outer class with a function display, again create another class inside the outer class named inner with a function called display and call the two functions in the main class.

## Lab Exercise 13 (2 Hours)

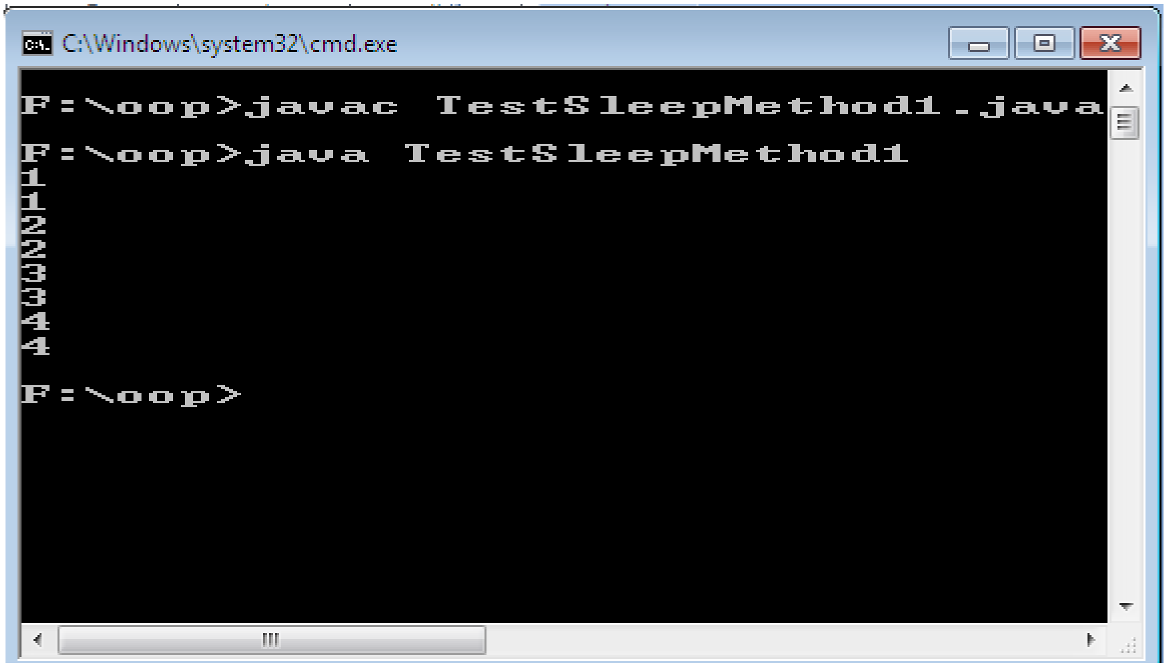
**Aim**: Program to demonstrate multithreading programming

**Problem Stmt:** WAP for sleep method in Thred

**Source Code:**

```
class TestSleepMethod1 extends  Thread{
 public void run(){
  for(int i=1;i<5;i++)
{
    try{Thread.sleep(500);}

catch(InterruptedException e)
{System.out.println(e);}
    System.out.println(i);
  }
 }
 public static void main(String  args[]){
  TestSleepMethod1 t1=new
TestSleepMethod1();
  TestSleepMethod1 t2=new
TestSleepMethod1();
  t1.start();
  t2.start();
 }
}
```

**Output:**



```
F:\oop>javac TestSleepMethod1.java

F:\oop>java TestSleepMethod1
1
1
2
2
3
3
4
4

F:\oop>
```

**Exercise:**

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.

WAP to illustrate priority of a Thread

WAP to illustrate garbage Collection

## Lab Exercise 14 (2 Hours)

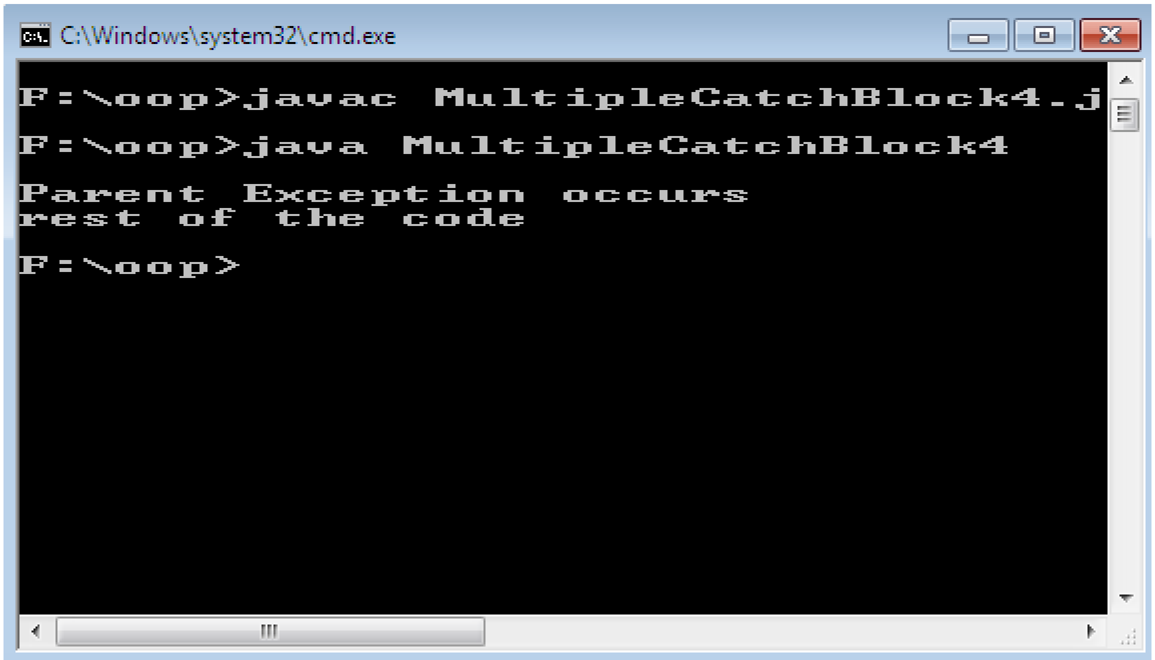
**Aim**: Program to demonstrate use of exception handling

**Problem Stmt:** WAP for Null Pointer Exception

**Source Code:**

```
public class MultipleCatchBlock4
{       public static void main(String[] args) {

             try{

              String s=null;

              System.out.println(s.length());

              }

             catch(ArithmeticException  e)

               {  System.out.println("Arithmetic  Exception occurs");  }

             catch(ArrayIndexOutOfBoundsException e)

               {

System.out.println("ArrayIndexOutOfBounds  Exception occurs");

               }

             catch(Exception e)

               {

   System.out.println("\nParent   Exception occurs");

               }

           System.out.println("rest  of the code");

   }

}
```

## Output:



```
F:\oop>javac MultipleCatchBlock4.j
F:\oop>java MultipleCatchBlock4
Parent Exception occurs
rest of the code
F:\oop>
```

## Exercise:

Write a 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 Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

WAP to illustrate Finally Block

WAP to illustrate Throws Keyword

## Lab Exercise 15 (2 Hours)

**Aim**: Program to demonstrate Applet Programming

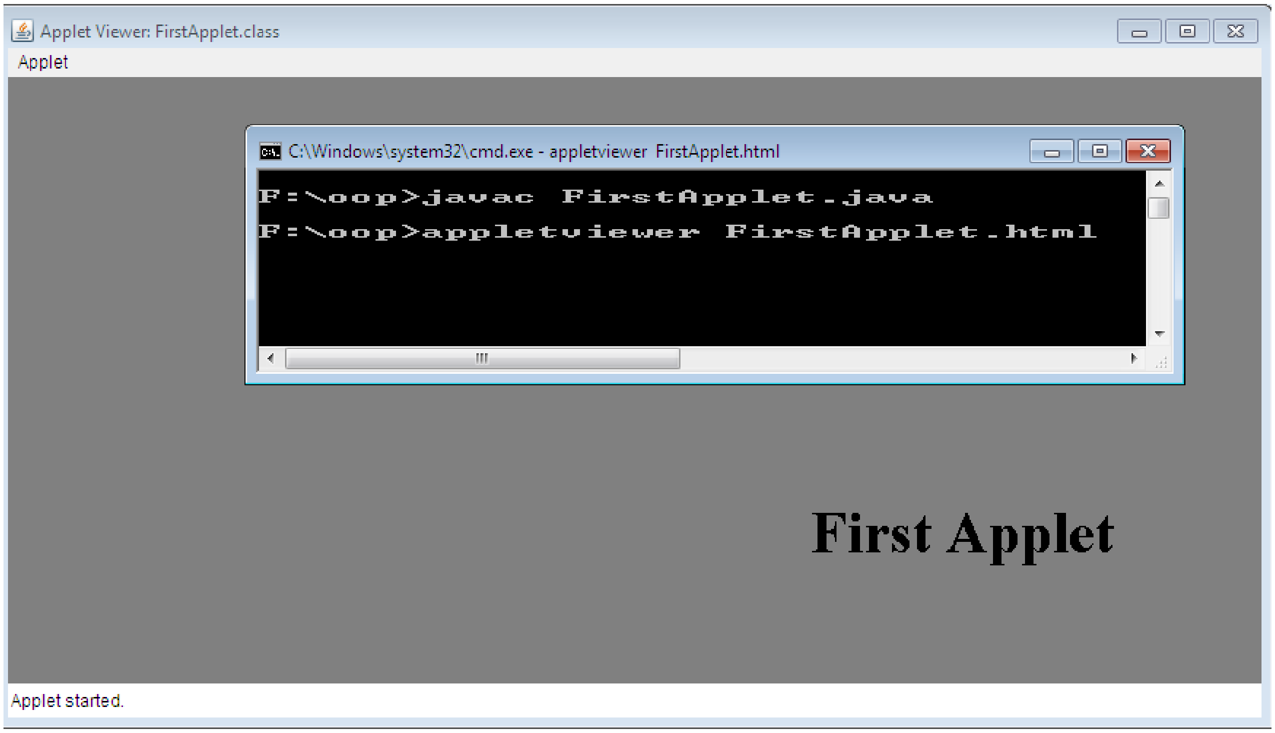
**Problem Stmt:**WAP to display Msg using drawstring method in Applet

**Source Code:**

```
//FirstApplet.java
import java.awt.*;
import java.applet.*;
public class FirstApplet extends Applet
{
public void init()
{
}
public void paint(Graphics g)
{
setBackground(Color.gray);
Font f1=new Font("Times New Roman",Font.BOLD,40);
g.setFont(f1);
g.drawString("First Applet",550,325);
}
}
```

```
//FirstApplet.html
<html>
<body>
<Applet code="FirstApplet.class" width="1000" height="600">
</Applet>
</body>
</html>
```

**Output:**


Applet Viewer: FirstApplet.class
Applet
C:\Windows\system32\cmd.exe - appletviewer FirstApplet.html

```
F:\oop>javac FirstApplet.java
F:\oop>appletviewer FirstApplet.html
```

First Applet

Applet started.

**Exercise:**

WAP for painting in Applet

WAP ti display image in Applet

## Lab Exercise 16 (2 Hours)

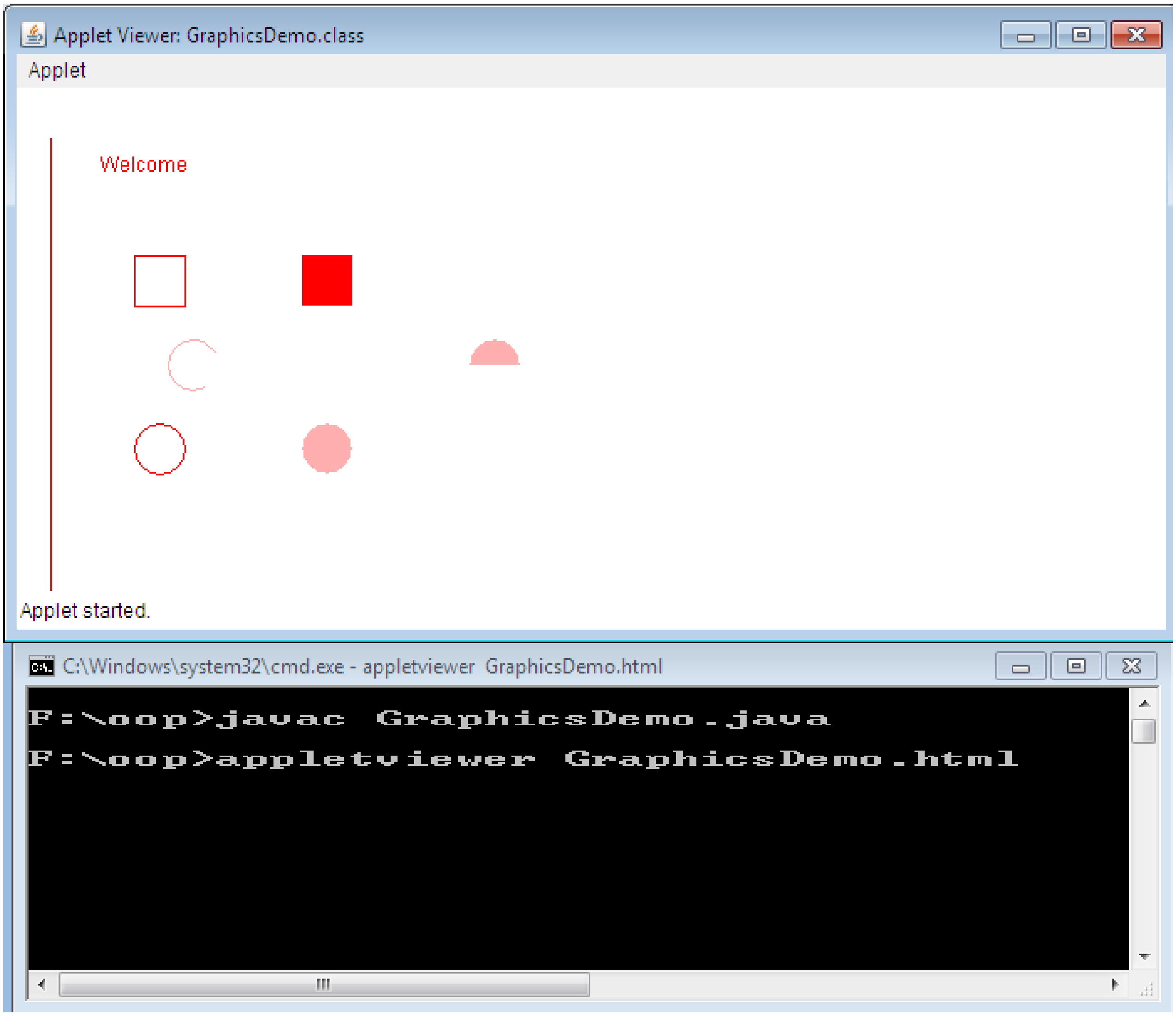
**Aim**: Program to demonstrate use of Graphics Programming

**Problem Stmt:** WAP to display different shapes and msg using Applet

**Source Code:**

```
import java.applet.Applet;
import java.awt.*;
public class GraphicsDemo extends Applet{
public void paint(Graphics g){
g.setColor(Color.red);
g.drawString("Welcome",50, 50);
g.drawLine(20,30,20,300);
g.drawRect(70,100,30,30);
g.fillRect(170,100,30,30);
g.drawOval(70,200,30,30);
 g.setColor(Color.pink);
g.fillOval(170,200,30,30);
g.drawArc(90,150,30,30,30,270);
g.fillArc(270,150,30,30,0,180);

}
}
```

## Output:


Applet Viewer: GraphicsDemo.class
Applet
C:\Windows\system32\cmd.exe - appletviewer GraphicsDemo.html

Welcome

Applet started.

```
F:\oop>javac GraphicsDemo.java
F:\oop>appletviewer GraphicsDemo.html
```

## Exercise:

WAP for digital clock in Applet

WAP for draw Smiley face

## Lab Exercise 17 (2 Hours)

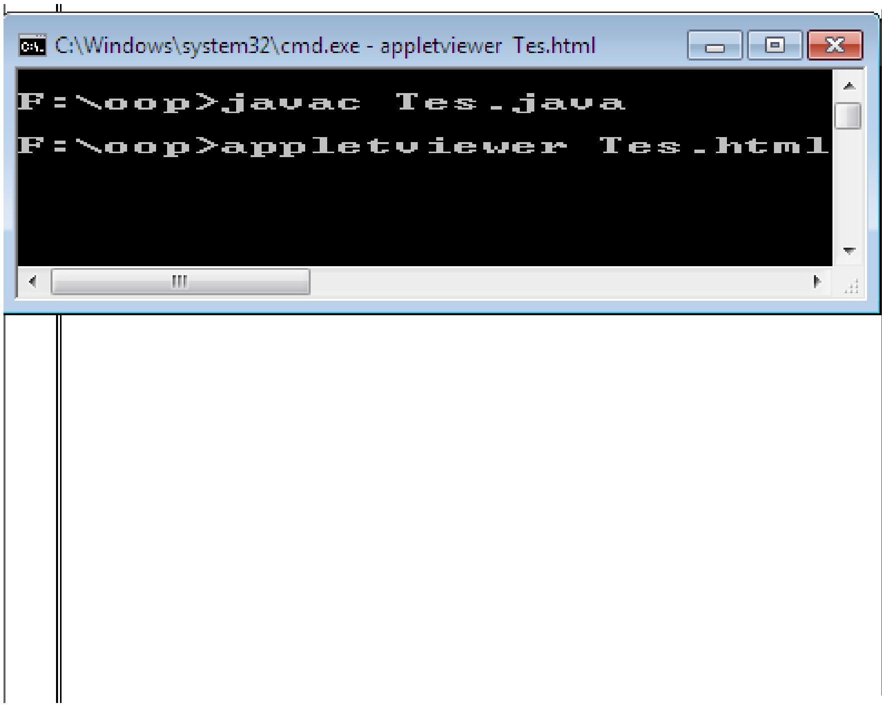
**Aim**: Program ti illustrate use of AWT

**Problem Stmt:** WAP to create Student Input Form in Applet

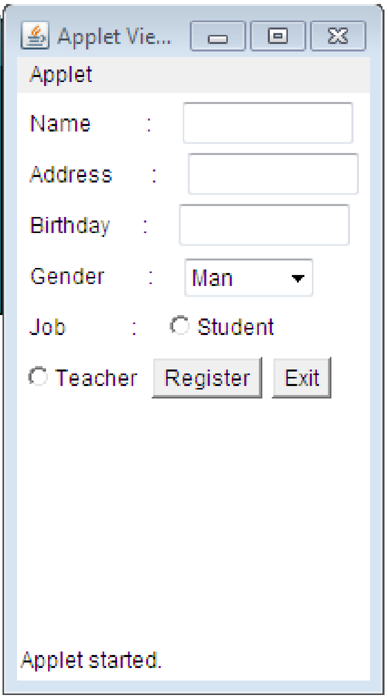
**Source Code:**

```
//Tes.java
import java.awt.*;
public class Tes extends java.applet.Applet
{
    public void init()
    {
         setLayout(new FlowLayout(FlowLayout.LEFT));
         add(new Label("Name           :"));
        add(new TextField(10));
        add(new Label("Address        :"));
        add(new TextField(10));
        add(new Label("Birthday       :"));
        add(new TextField(10));
         add(new Label("Gender        :"));
        Choice gender = new Choice();
        gender.addItem("Man");
        gender.addItem("Woman");
        Component add = add(gender);
        add(new Label("Job          :"));
        CheckboxGroup job = new CheckboxGroup();
        add(new Checkbox("Student", job, false));
        add(new Checkbox("Teacher", job, false));
        add(new Button("Register"));
        add(new Button("Exit"));
        }
}

//Tes.html
<html>
<head><title>Register</title></head>
<body>
<applet code="Tes.class" width=230 height=300></applet>
</body>
</html>
```

## Output:



```
F:\oop>javac Tes.java
F:\oop>appletviewer Tes.html
```



Name :

Address :

Birthday :

Gender : Man

Job : ○ Student

○ Teacher

Applet started.

## Exercise:

WAP to illustrate Mouse Listener Event

WAP to create login form for User

WAP free hand drawing Tool

Write a java program to create an abstract class named Shape that contain an empty method named numberOfSides(). Provide three classes named Trapezoid, Triangle and Hexagon such that each one of the classes extends the class Shape. Each one of the class contains only the method numberOfSides() that shows the number of sides in the given geometrical figures.

## Lab Exercise 18 (2 Hours)

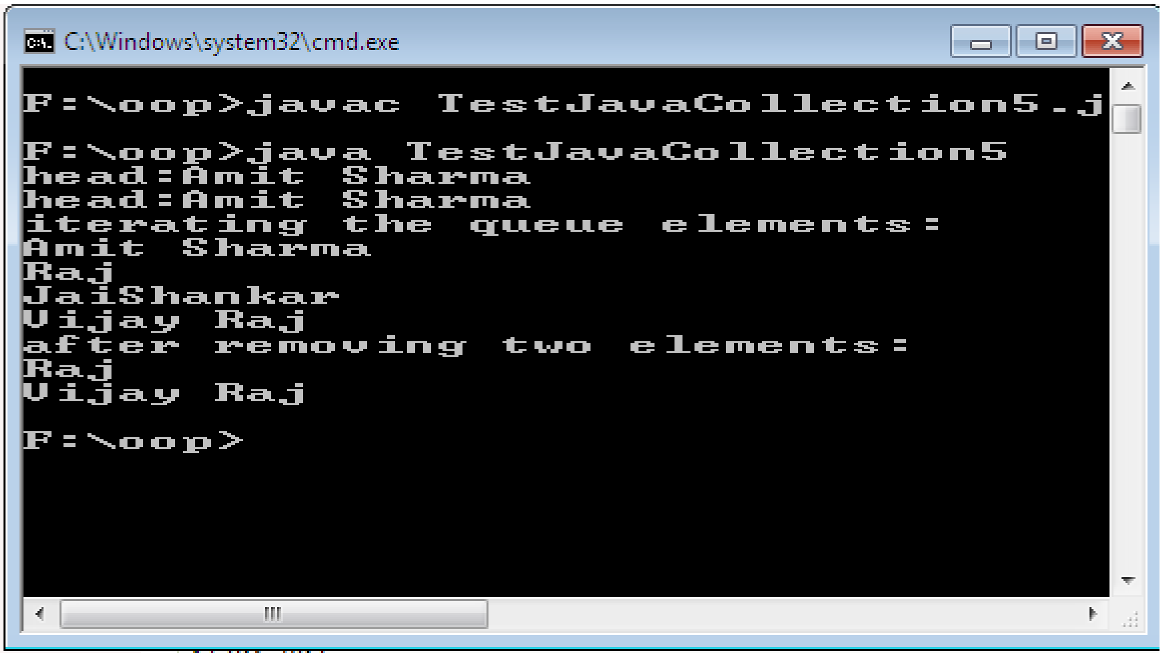
**Aim**: Program to demonstrate use of collections

**Problem Stmt:** WAP for add and delete elements in queue Collections

**Source Code:**

```
//priority Queue

import java.util.*;

public class TestJavaCollection5{

public static void main(String args[]){

PriorityQueue<String> queue=new
PriorityQueue<String>();

queue.add("Amit Sharma");

queue.add("Vijay Raj");

queue.add("JaiShankar");

queue.add("Raj");

System.out.println("head:"+queue.element());

System.out.println("head:"+queue.peek());

System.out.println("iterating the queue elements:");

Iterator itr=queue.iterator();

while(itr.hasNext()){

System.out.println(itr.next());

}
queue.remove();

queue.poll();

System.out.println("after  removing  two elements:");

Iterator<String>
itr2=queue.iterator();

while(itr2.hasNext()){

System.out.println(itr2.next());
}  }  }
```

**Output:**



```
F:\oop>javac TestJavaCollection5.j

F:\oop>java TestJavaCollection5
head:Amit Sharma
head:Amit Sharma
iterating the queue elements:
Amit Sharma
Raj
JaiShankar
Vijay Raj
after removing two elements:
Raj
Vijay Raj

F:\oop>
```

**Exercise:**

WAP to illustrate  use of TreeSet Class

WAp to illustrate  HAshMap class

## Lab Exercise 19 (2 Hours)

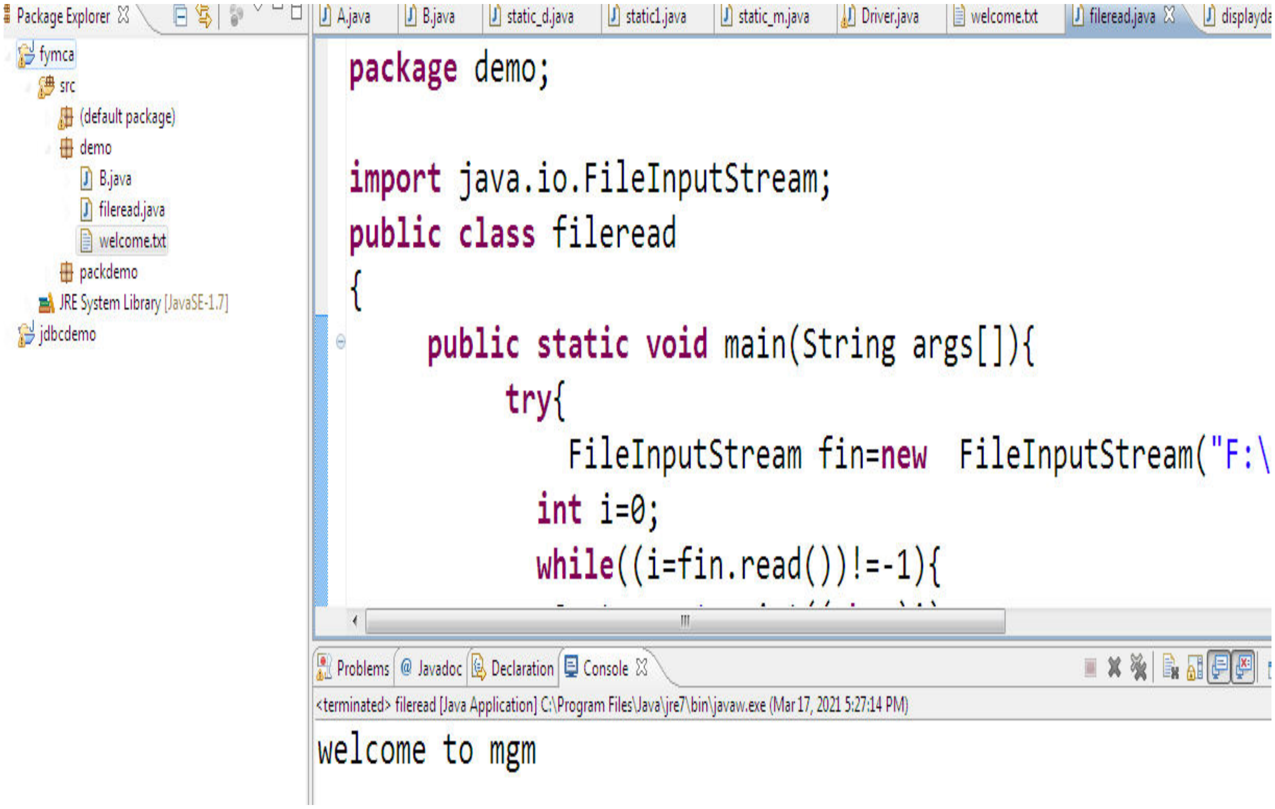
**Aim**: Program to demonstrate use of File IO

**Problem Stmt:** WAP to read character from text file (eg. F: \\abc.txt)

**Source Code:**

```
package demo;

import java.io.FileInputStream;

public class fileread

{

    public static void main(String args[]){

        try{

           FileInputStream fin=new
FileInputStream("F:\\oop\\fymca\\src\\demo\\welcome.txt");

           int i=0;

           while((i=fin.read())!=-1){

            System.out.print((char)i);

           }

           fin.close();

        }catch(Exception e){System.out.println(e);}

       }

      }
```

## Output:



```
package demo;

import java.io.FileInputStream;
public class fileread
{
        public static void main(String args[]){
              try{
                   FileInputStream fin=new  FileInputStream("F:\
                int i=0;
                while((i=fin.read())!=-1){
```

```
<terminated> fileread [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Mar 17, 2021 5:27:14 PM)
welcome to mgm
```

## Exercise:

WAP for string writer

## Lab Exercise 20 (2 Hours)

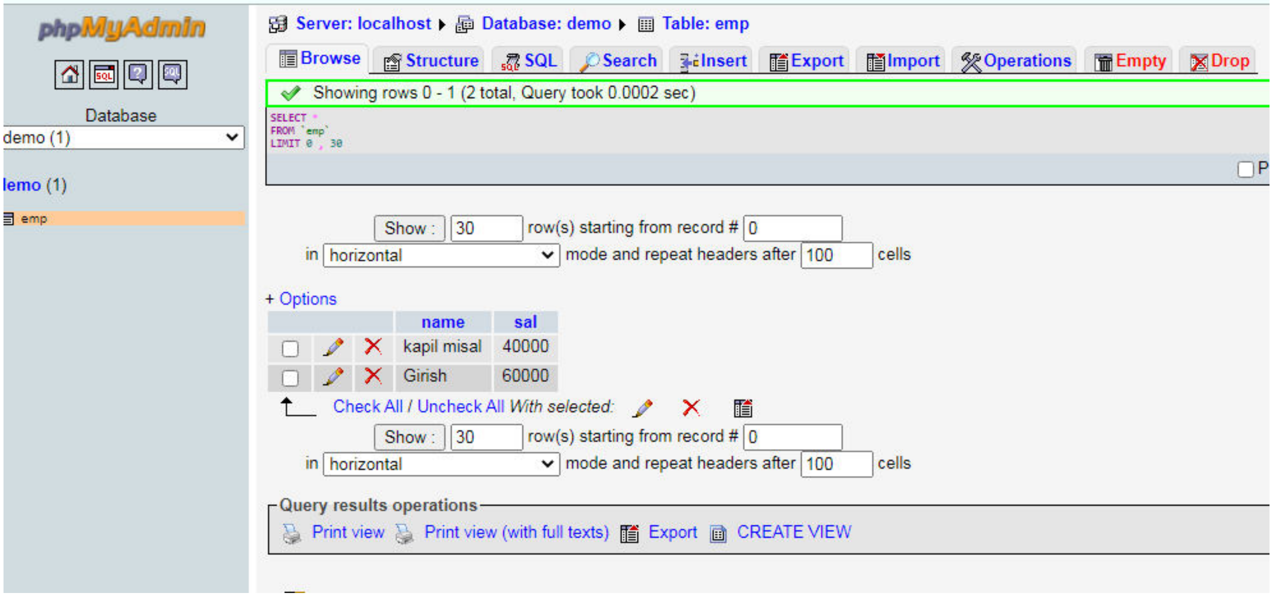
**Aim**: Program to demonstrate use of JDBC

**Problem Stmt:** WAP to display data from MYSQL Database

**Source Code:**

```
//jdbcdemo.java

package jdbcdemo;

import java.sql.*;

class displaydata

{

public static void main(String  args[])

{

try{

Class.forName("com.mysql.jdbc.Driver");

Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/demo",
"root","");

System.out.print("Database is connected !\n");

//here demo is database name, root is username and password

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select * from emp");

while(rs.next())

System.out.println(rs.getString(1)+"   "+rs.getInt(2));

con.close();

}catch(Exception e){ System.out.println(e);}

}  }
```

## Output:

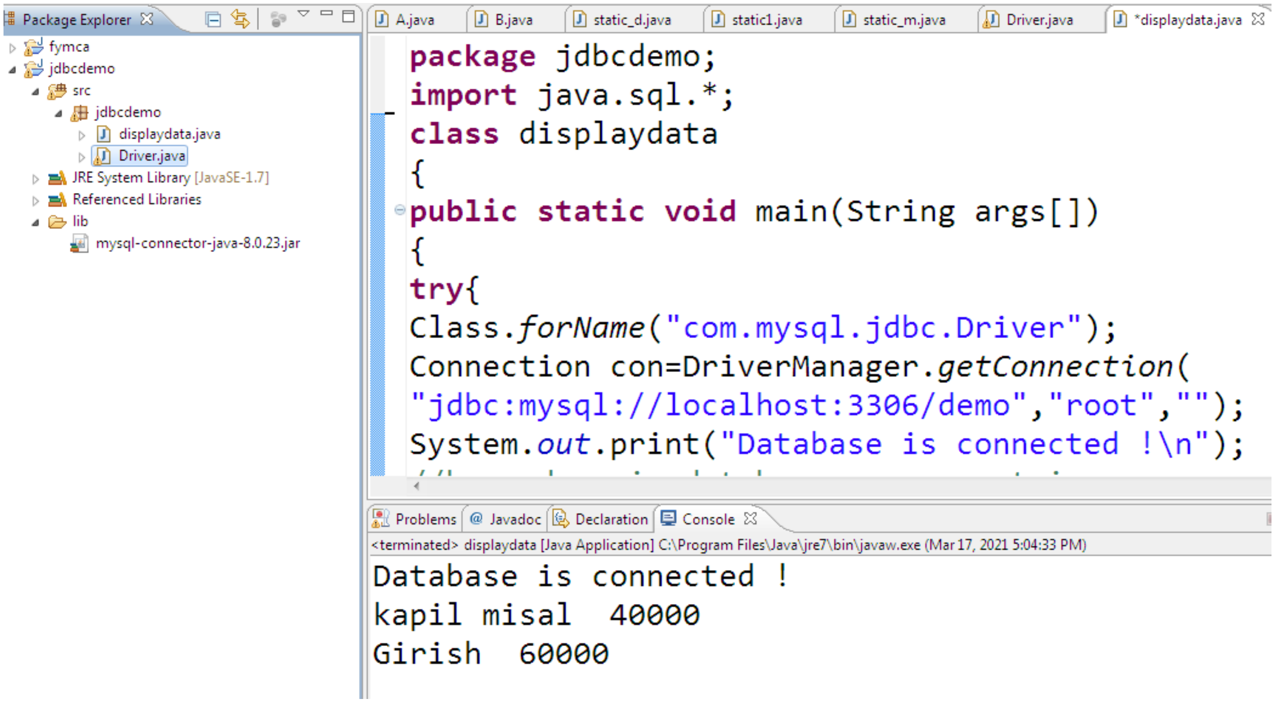


✓ Showing rows 0 - 1 (2 total, Query took 0.0002 sec)

```
SELECT *
FROM `emp`
LIMIT 0 , 30
```

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

| | | | name | sal |
|---|---|---|---|---|
| □ | ✎ | ✕ | kapil misal | 40000 |
| □ | ✎ | ✕ | Girish | 60000 |



```
package jdbcdemo;
import java.sql.*;
class displaydata
{
public static void main(String args[])
{
try{
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/demo","root","");
System.out.print("Database is connected !\n");
```

```
<terminated> displaydata [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Mar 17, 2021 5:04:33 PM)
Database is connected !
kapil misal  40000
Girish  60000
```

## Exercise:

WAP to store and retrieve student photo and details from DB

WAP to store and retrieve a file from DB

## Quiz on the subject:

Quiz should be conducted on tips in the laboratory, recent trends and subject knowledge of the subject. The quiz questions should be formulated such that questions are normally from the scope outside of the books. However twisted questions and self-formulated questions by the faculty can be asked but correctness of it is necessarily to be thoroughly checked before the conduction of the quiz.

## Conduction of Viva-Voce Examinations:

Teacher should oral exams of the students with full preparation. Normally, the objective questions with guess are to be avoided. To make it meaningful, the questions should be such that depth of the students in the subject is tested Oral examinations are to be conducted in co-cordial environment amongst the teachers taking the examination. Teachers taking such examinations should not have ill thoughts about each other and courtesies should be offered to each other in case of difference of opinion, which should be critically suppressed in front of the students..

## Submission:

Document Standard:

| | |
|---|---|
| A] Page Size | A4-Size |
| B] Running text | Justified text |
| C] Spacing | 1.5 line |
| D] Page Layout and Margins (Dimensions in Cms) | |

## Evaluation and marking system:

Basic honesty in the evaluation and marking system is absolutely essential and in the process impartial nature of the evaluator is required in the examination system to become popular amongst the students. It is a wrong approach or concept to award the students by way of easy marking to get cheap popularity among the students to which they do not deserve. It is a primary responsibility of the teacher that right students who are really putting up lot of hard work with right kind of intelligence are correctly awarded.

The marking patterns should be justifiable to the students without any ambiguity and teacher should see that students are faced with unjust circumstances.