Slip No 1

1.Write a Program to print all Prime numbers in an array of ‘n’ elements.

```
import java.util.Scanner;
public class PrimeNumbers{
public static void main (String[] args){
int[] array = new int [5];
Scanner in = new Scanner (System.in);

System.out.println("Enter the elements of the array: ");
for(int i=0; i<5; i++)
{
array[i] = in.nextInt();
}
```

2. Define an abstract class Staff with protected members id and name. Define a parameterized constructor. Define one subclass OfficeStaff with member department. Create n objects of OfficeStaff and display all details.

```
import java.io.IOException;

import java.io.InputStreamReader;

abstract class Staff{

String name,address;

}

class FullTimeStaff extends Staff{

String department;

double salary;

public void accept() throws IOException{

System.out.println("Enter the name, address, department and salary: ");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

name=br.readLine();

address=br.readLine();

department=br.readLine();

salary=Double.parseDouble(br.readLine());

}

public void display(){
```

```
System.out.println("Name: "+name);
System.out.println("Address: "+address);
System.out.println("Department: "+department);
System.out.println("Salary: "+salary);
System.out.println("----------------------");
}
}
class PartTimeStaff extends Staff{
int hours, rate;
public void accept() throws IOException{
System.out.println("Enter the name, address, No of working hours and rate per hour: ");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
name=br.readLine();
address=br.readLine();
hours=Integer.parseInt(br.readLine());
rate=Integer.parseInt(br.readLine());
}
public void display(){
System.out.println("Name: "+name);
System.out.println("Address: "+address);
System.out.println("No of Working Hours: "+hours);
System.out.println("Rate per hour: "+rate);
System.out.println("----------------------");
}
}
public class sb1 {
public static void main(String [] args) throws IOException{
int i;
System.out.println("Select Any One: ");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
```

```
System.out.println("1.Full Time Staff");
System.out.println("2.Part Time Satff");
int ch=Integer.parseInt(br.readLine());
switch(ch){
case 1:
System.out.println("Enter the number of Full Time Staff: ");
int n=Integer.parseInt(br.readLine());
FullTimeStaff [] l=new FullTimeStaff[n];
for(i=0;i<n;i++){
l[i]=new FullTimeStaff();
l[i].accept();
}
for(i=0;i<n;i++){
l[i].display();
}
break;
case 2:
System.out.println("Enter the number of Part Time Staff: ");
int m=Integer.parseInt(br.readLine());
PartTimeStaff [] h=new PartTimeStaff[m];
for(i=0;i<m;i++){
h[i]=new PartTimeStaff();
h[i].accept();
}
for(i=0;i<m;i++){
h[i].display();
}
break;
}
}
```

}

Output:

```
Select Any One:
1.Full Time Staff
2.Part Time Satff
1
Enter the number of Full Time Staff:
2
Enter the name, address, department and salary:
Rohit
Pune
AB
19000
Enter the name, address, department and salary:
Karan
Mumbai
BA
18000
Name: Rohit
Address: Pune
Department: AB
Salary: 19000.0
Name: Karan
Address: Mumbai
Department: BA
Salary: 18000.0
```

Slip No 2

1.Write a program to read the First Name and Last Name of a person, his weight and height using command line arguments. Calculate the BMI Index which is defined as the individual's body massdivided by the square of their height.
(Hint : BMI = Wts. In kgs / $(ht)^2$)

```
import java.util.Scanner;

public class Exercise6 {


   public static void main(String[] Strings) {


      Scanner input = new Scanner(System.in);


      System.out.print("Input weight in pounds: ");

      double weight = input.nextDouble();


      System.out.print("Input height in inches: ");

      double inches = input.nextDouble();


      double BMI = weight * 0.45359237 / (inches * 0.0254 * inches * 0.0254);

      System.out.print("Body Mass Index is " + BMI+"\n");

   }

}
```

Input weight in pounds: 452

Input height in inches: 72

Body Mass Index is 61.30159143458721

2. Define a class CricketPlayer (name,no_of_innings,no_of_times_notout, totatruns, bat_avg). Create an array of n player objects .Calculate the batting average for each player using static method avg(). Define a static sort method which sorts the array on the basis of average. Display the player details in sorted order.

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

```
class Cricket {
String name;
int inning, tofnotout, totalruns;
float batavg;
public Cricket(){
name=null;
inning=0;
tofnotout=0;
totalruns=0;
batavg=0;
}
public void get() throws IOException{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter the name, no of innings, no of times not out, total runs: ");
name=br.readLine();
inning=Integer.parseInt(br.readLine());
tofnotout=Integer.parseInt(br.readLine());
totalruns=Integer.parseInt(br.readLine());
}
public void put(){
System.out.println("Name="+name);
System.out.println("no of innings="+inning);
System.out.println("no times notout="+tofnotout);
System.out.println("total runs="+totalruns);
System.out.println("bat avg="+batavg);
}
static void avg(int n, Cricket c[]){
try{
for(int i=0;i<n;i++){
c[i].batavg=c[i].totalruns/c[i].inning;
```

```
}
}catch(ArithmeticException e){
System.out.println("Invalid arg");

}
}
static void sort(int n, Cricket c[]){
String temp1;
int temp2,temp3,temp4;
float temp5;
for(int i=0;i<n;i++){
for(int j=i+1;j<n;j++){
if(c[i].batavg<c[j].batavg){
temp1=c[i].name;
c[i].name=c[j].name;
c[j].name=temp1;
temp2=c[i].inning;
c[i].inning=c[j].inning;
c[j].inning=temp2;
temp3=c[i].tofnotout;
c[i].tofnotout=c[j].tofnotout;
c[j].tofnotout=temp3;
temp4=c[i].totalruns;
c[i].totalruns=c[j].totalruns;
c[j].totalruns=temp4;
temp5=c[i].batavg;
c[i].batavg=c[j].batavg;
c[j].batavg=temp5;
}
}
```

```
}
}
}
public class a4sa1 {
public static void main(String args[])throws IOException{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter the limit:");
int n=Integer.parseInt(br.readLine());
Cricket c[]=new Cricket[n];
for(int i=0;i<n;i++){
c[i]=new Cricket();
c[i].get();
}
Cricket.avg(n,c);
Cricket.sort(n, c);
for(int i=0;i<n;i++){
c[i].put();
}
}
}
```

Output:

```
Enter the limit:
2
Enter the name, no of innings, no of times not out, total runs:
Virat
3
1
100
Enter the name, no of innings, no of times not out, total runs:
```

Rohit

2

1

120

Name=Rohit

no of innings=2

no times notout=1

total runs=120

bat avg=60.0

Name=Virat

no of innings=3

no times notout=1

total runs=100

bat avg=33.0

Slip No 3

1. Write a program to accept ‘n’ name of cities from the user and sort them in ascending order.

```
class SortCity
{
	public static void main(String arg[])
	{
		String name[]=new String[10];
		int l=arg.length;
		String temp;
			for(int i=0;i<l;i++)
			{
				name[i]=arg[i];
			}
```

```
                for(int j=0;j<l;j++)
                {
                        for(int k=j+1;k<l;k++)
                        {
                                if((name[j].compareTo(name[k]))<0)
                                {
                                        temp=name[j];
                                        name[j]=name[k];
                                        name[k]=temp;
                                }
                        }
                }
                System.out.println("Sorted City Are-");
                for(int i=0;i<l;i++)
                {
                        System.out.println(name[i]);
                }
        }
}

/*OUTPUT
D:\javapro>javac SortCity.java
D:\javapro>java SortCity  pune nagar solapur thane satara baramati
Sorted City Are-
thane
solapur
satara
pune
nagar
Baramati
```

2.Define a class patient (patient_name, patient_age, patient_oxy_level,patient_HRCT_report). Create an object of patient. Handle appropriate exception while patient oxygen level less than 95% and HRCT scan report greater than 10, then throw user defined Exception “Patient is Covid Positive(+) and Need to Hospitalized” otherwise display its information.

```
import java.util.*;

class Patient

{

String name;

int age;

int oxylevel;

int HRCTreport;

patient(String name, int age, int oxylevel, int HRCTreport)

{

this.name = name;

this.age = age;

this.oxylevel = oxylevel;

this.HRCTreport = HRCTreport;

}

}

public class Main extends Exception

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("How many patient you want insert:");

int number = sc.nextInt();

patient[] ob = new patient[number];

for(int j=0; j<number; j++)

{

System.out.println("Enter Name ");
```

```
String name = sc.next();
System.out.println("Enter Age ");
int age = sc.nextInt();
System.out.println("Enter oxygen level");
int oxylevel = sc.nextInt();
System.out.println("Enter HRCT report");
int HRCTreport = sc.nextInt();
ob[j] = new patient(name, age, oxylevel, HRCTreport);
}

for(int j=0; j<number; j++)
{
if(ob[j].oxylevel < 95 && ob[j].HRCTreport > 10)
try
{

throw new NullPointerException("\n");
}
catch(Exception e)
{
System.out.println("Patient is Covid Positive(+) and Need to Hospitalized\

n");

}
else
{
System.out.println("name: "+ob[j].name);
System.out.println("age " + ob[j].age);
System.out.println("oxygen level " +ob[j].oxylevel);
```

```
System.out.println("HRCT report " + ob[j].HRCTreport);
System.out.println("\n");
}
}
}
}
```

Output:

```
How many patient you want insert:
2
Enter Name
abc
Enter Age
109
Enter oxygen level
70
Enter HRCT report
10
Enter Name
xyz
Enter Age
111
Enter oxygen level
60
Enter HRCT report
9
name: abc
age 109
oxygen level 70
```

HRCT report 10

name: xyz

age 111

oxygen level 60

HRCT report 9

Slip No 4

1. Write a program to print an array after changing the rows and columns of a given two-dimensional array.

```
import java.util.Scanner;
public class Solution {
	public static void main(String[] args) {
		int[][] twodm = {
				{10, 20, 30},
				{40, 50, 60}
		};
		System.out.print("Original Array:\n");
		print_array(twodm);
		System.out.print("After changing the rows and columns of the said array:");
		transpose(twodm);
		}

	private static void transpose(int[][] twodm) {

		int[][] newtwodm = new int[twodm[0].length][twodm.length];

		for (int i = 0; i < twodm.length; i++) {
			for (int j = 0; j < twodm[0].length; j++) {
				newtwodm[j][i] = twodm[i][j];
```

```
                }
            }

            print_array(newtwodm);
        }
        private static void print_array(int[][] twodm) {
            for (int i = 0; i < twodm.length; i++) {
                for (int j = 0; j < twodm[0].length; j++) {
                    System.out.print(twodm[i][j] + " ");
                }
                System.out.println();
            }

        }
}
```

2.Write a Java program to design a screen using Awt that will take a user name and password. If the user name and password are not same, raise an exception with appropriate message. User can have 3 login chances only. Use clear 7 button to clear the text boxes.

```
import java.awt.*;
import java.awt.event.*;
class InvalidPasswordException extends Exception
{
InvalidPasswordException()
{
System.out.println(" User name and Password is not same");
}
}
public class PasswordDemo extends Frame implements ActionListener
{
```

```
Label uname,upass;
TextField nametext;
TextField passtext,msg;
Button login,Clear;
Panel p;
int attempt=0;
char c= ‘ * ‘ ;

public void login()
{
p=new Panel();
uname=new Label(“Use Name: ” ,Label.CENTER);
upass=new Label (“Password: “,Label.RIGHT);

nametext=new TextField(20);
passtext =new TextField(20);
passtext.setEchoChar(c);
msg=new TextField(10);
msg.setEditable(false);

login=new Button(“Login”);
Clear=new Button(“Clear”);
login.addActionListener(this);
Clear.addActionListener(this);

p.add(uname);
p.add(nametext);
p.add(upass);
p.add(passtext);
p.add(login);
```

```
p.add(Clear);
p.add(msg);
add(p);

setTitle(“Login “);
setSize(290,200);
setResizable(false);
setVisible(true);
}

public void actionPerformed(ActionEvent ae)
{
Button btn=(Button)(ae.getSource());
if(attempt<3)
{
if((btn.getLabel())==”Clear”)
{
nametext.setText(“”);
passtext.setText(“”);
}
if((btn.getLabel()).equals(“Login”))
{
try
{
String user=nametext.getText();
String upass=passtext.getText();

if(user.compareTo(upass)==0)
{
msg.setText(“Valid”);
```

```
System.out.println(“Username is valid”);
}
else
{
throw new InvalidPasswordException();
}
}
catch(Exception e)
{
msg.setText(“Error”);
}
attempt++;
}
}
else
{
System.out.println(“you are using 3 attempt”);
System.exit(0);
}
}
public static void main(String args[])
{
PasswordDemo pd=new PasswordDemo();
pd.login();
}
}
```

Slip No 5

1.Write a program for multilevel inheritance such that Country is inherited from Continent. State is inherited from Country. Display the place, State, Country and Continent.

```
import java.io.BufferedReader;
import java.io.IOException;
class Continent
{
String con;
InputStreamReader i = new InputStreamReader(System.in);
BufferedReader r = new BufferedReader(i);
void con_input() throws IOException
{
System.out.println("Enter Continent Name: ");
con = r.readLine();
}
}
class Country extends Continent
{
String cou ;
void cou_input() throws IOException
{
System.out.println("Enter Country Name: ");
cou = r.readLine();
}
}
class State extends Country
{
String sta;
void sta_input() throws IOException
{
System.out.println("Enter State Name: ");
sta = r.readLine();
}
}
class Main extends State
{
String pla;
void pla_input()throws IOException
{
System.out.println("Enter Place Name : ");
pla = r.readLine();
}
public static void main( String argsp[] )throws IOException
{
Main s = new Main();
s.con_input();
```

```
s.cou_input();
s.sta_input();
s.pla_input();
System.out.println("\n\nContinent: "+s.con);
System.out.println("Country: "+s.cou);
System.out.println("State: "+s.sta);
System.out.println("Place :" + s.pla);
}
}
```

Output:

```
Enter Continent Name:
Asia
Enter Country Name:
India
Enter State Name:
Maharashtra
Enter Place Name :
Pune
```

```
Continent: Asia
Country: India
State: Maharashtra
Place :Pune
```

2. Write a menu driven program to perform the following operations on multidimensional array ie matrices :

- Addition
- Multiplication
- Exit

```
import java.util.Arrays;

import java.util.Scanner;

public class Matrix {

  // main method

  public static void main(String[] args) {

    // Scanner class object

    Scanner scan = new Scanner(System.in);
```

```
// declare two matrix
int a[][] = { { 5, 6, 7 }, { 8, 9, 10 }, { 3, 1, 2 } };
int b[][] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };
// create third matrix
int c[][] = new int[3][3];
// display both matrix
System.out.println("A = " + Arrays.deepToString(a));
System.out.println("B = " + Arrays.deepToString(b));
// variable to take choice
int choice;
// menu-driven
do {
  // menu to choose the operation
  System.out.println("\nChoose the matrix operation,");
  System.out.println("----------------------------");
  System.out.println("1. Addition");
  System.out.println("2. Subtraction");
  System.out.println("3. Multiplication");
  System.out.println("4. Transpose");
  System.out.println("5. Exit");
  System.out.println("----------------------------");
  System.out.print("Enter your choice: ");
  choice = scan.nextInt();
  switch (choice) {
  case 1:
    c = add(a, b);
    System.out.println("Sum of matrix: ");
    System.out.println(Arrays.deepToString(c));
    break;
  case 2:
```

```
      c = subtract(a, b);
      System.out.println("Subtraction of matrix: ");
      System.out.println(Arrays.deepToString(c));
      break;
    case 3:
      c = multiply(a, b);
      System.out.println("Multiplication of matrix: ");
      System.out.println(Arrays.deepToString(c));
      break;
    case 4:
      System.out.println("Transpose of the first matrix: ");
      c = transpose(a);
      System.out.println(Arrays.deepToString(c));
      System.out.println("Transpose of the second matrix: ");
      c = transpose(b);
      System.out.println(Arrays.deepToString(c));
      break;
    case 5:
      System.out.println("Thank You.");
      return;
    default:
      System.out.println("Invalid input.");
      System.out.println("Please enter the correct input.");
    }
  } while (true);
}

// method to perform matrix addition and
// return resultant matrix
public static int[][] add(int[][] a, int[][] b) {
```

```
  // calculate row and column size of anyone matrix
  int row = a.length;
  int column = a[0].length;
  // declare a matrix to store resultant value
  int sum[][] = new int[row][column];
  // calculate sum of two matrices
  for (int i = 0; i < row; i++) {
    for (int j = 0; j < column; j++) {
      sum[i][j] = a[i][j] + b[i][j];
    }
  }
  // return resultant matrix
  return sum;
}
// method to perform matrix subtraction and
// return resultant matrix
public static int[][] subtract(int[][] a, int[][] b) {
  // calculate row and column size of anyone matrix
  int row = a.length;
  int column = a[0].length;
  // declare a matrix to store resultant value
  int sub[][] = new int[row][column];
  // calculate sum of two matrices
  for (int i = 0; i < row; i++) {
    for (int j = 0; j < column; j++) {
      sub[i][j] = a[i][j] - b[i][j];
    }
  }
```

```
  // return resultant matrix
  return sub;
}
// method to perform matrix multiplication and
// return resultant matrix
// passed matrices can be square or non-square matrix
public static int[][] multiply(int[][] a, int[][] b) {

  // find row size of first matrix
  int row = a.length;
  // find column size of second matrix
  int column = b[0].length;

  // declare new matrix to store result
  int product[][] = new int[row][column];

  // find product of both matrices
  // outer loop up to row of A
  for (int i = 0; i < row; i++) {
    // inner-1 loop utp0 column of B
    for (int j = 0; j < column; j++) {
      // assign 0 to the current element
      product[i][j] = 0;

      // inner-2 loop up to A[0].length
      for (int k = 0; k < a[0].length; k++) {
        product[i][j] += a[i][k] * b[k][j];
      }
    }
  }
```

```
    return product;
  }

  // method to find transpose of a matrix
  public static int[][] transpose(int[][] a) {

    // calculate row and column size
    int row = a.length;
    int column = a[0].length;

    // declare a matrix to store resultant
    int temp[][] = new int[row][column];

    // calculate transpose of matrix
    // outer loop for row
    for (int i = 0; i < row; i++) {
      // inner loop for column
      for (int j = 0; j < column; j++) {
        // formula
        temp[i][j] = a[j][i];
      }
    }

    // return resultant matrix
    return temp;
  }

}
```

Slip No 6

1. Write a program to display the Employee(Empid, Empname, Empdesignation, Empsal) information using toString().

```
import java.util.Scanner;
public class Employee {
 int empid;
 String name;
 float salary;
 public void getInput() {
   Scanner in = new Scanner(System.in);
  System.out.print("Enter the empid :: ");
  empid = in.nextInt();
  System.out.print("Enter the name :: ");
  name = in.next();
  System.out.print("Enter the salary :: ");
  salary = in.nextFloat();
 }
 public void display() {
  System.out.println("Employee id = " + empid);
  System.out.println("Employee name = " + name);
  System.out.println("Employee salary = " + salary);
 }
  public static void main(String[] args) {
   Employee e[] = new Employee[5];
    for(int i=0; i<5; i++) {
     e[i] = new Employee();
   e[i].getInput();
  }
    System.out.println("**** Data Entered as below ****");
```

```
  for(int i=0; i<5; i++) {
      e[i].display();
  }
 }
}
```

OUTPUT

```
C:\>javac Employee.java
C:\>java Employee
Enter the empid :: 101
Enter the name :: Smith
Enter the salary :: 30250.0
Enter the empid :: 103
```

2. Create an abstract class “order” having members id, description. Create two subclasses “PurchaseOrder” and “Sales Order” having members customer name and Vendor name respectively. Definemethods accept and display in all cases. Create 3 objects each of Purchase Order and Sales Order and accept and display details.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
abstract class Order{
String id,description;
}
class PurchaseOrder extends Order{
String Customername,Vendorname;
public void accept() throws IOException{
System.out.println("Enter the id,description,names of customers and vendors: ");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
id=br.readLine();
description=br.readLine();
```

```
Customername=br.readLine();
Vendorname=br.readLine();
}
public void display(){
System.out.println("id: "+id);
System.out.println("Description: "+description);
System.out.println("Customername: "+Customername);
System.out.println("Vendorname: "+Vendorname);
System.out.println("----------------------");
}
}
class SalesOrder extends Order{
String Customername,Vendorname;
public void accept() throws IOException{
System.out.println("Enter the id,description,names of customers and vendors: ");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
id=br.readLine();
description=br.readLine();
Customername=br.readLine();
Vendorname=br.readLine();
}
public void display(){
System.out.println("id: "+id);
System.out.println("Description: "+description);
System.out.println("Customername: "+Customername);
System.out.println("Vendorname: "+Vendorname);
System.out.println("----------------------");
}
}
public class Main {
```

```
public static void main(String [] args) throws IOException{

int i;
System.out.println("Select Any One: ");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("1.Purchase Order");
System.out.println("2.Sales Order");
int ch=Integer.parseInt(br.readLine());
switch(ch){
case 1:
System.out.println("Enter the number of purchase Orders: ");
int n=Integer.parseInt(br.readLine());
PurchaseOrder [] l=new PurchaseOrder[n];
for(i=0;i<n;i++){
l[i]=new PurchaseOrder();
l[i].accept();
}
for(i=0;i<n;i++){
l[i].display();
System.out.println ("Object is created");
}
break;
case 2:
System.out.println("Enter the number of sales orders: ");
int m=Integer.parseInt(br.readLine());
SalesOrder [] h=new SalesOrder[m];
for(i=0;i<m;i++){
h[i]=new SalesOrder();
h[i].accept();
}
```

```
for(i=0;i<m;i++){
h[i].display();
System.out.println(" Object is created ");
}
break;

}
}
}
```

Output:

```
Select Any One:
1.Purchase Order
2.Sales Order
1
Enter the number of purchase Orders:
2
Enter the id,description,names of customers and vendors:
1
Soap
Rahul
Gajni
Enter the id,description,names of customers and vendors:
2

Handwash
prince
abhishek
id: 1
```

Description: Soap

Customername: Rahul

Vendorname: Gajni

Object is created

id: 2

Description: Handwash

Customername: prince

Vendorname: abhishek

Slip No 7

1. Design a class for Bank. Bank Class should support following operations;

a. Deposit a certain amount into an account

b. Withdraw a certain amount from an account

c. Return a Balance value specifying the amount with details

```
import java.util.Scanner;
class BankDetails {
    private String accno;
    private String name;
    private String acc_type;
    private long balance;
    Scanner sc = new Scanner(System.in);
    //method to open new account
    public void openAccount() {
        System.out.print("Enter Account No: ");
        accno = sc.next();
        System.out.print("Enter Account type: ");
```

```
    acc_type = sc.next();
    System.out.print("Enter Name: ");
    name = sc.next();
    System.out.print("Enter Balance: ");
    balance = sc.nextLong();
}
//method to display account details
public void showAccount() {
    System.out.println("Name of account holder: " + name);
    System.out.println("Account no.: " + accno);
    System.out.println("Account type: " + acc_type);
    System.out.println("Balance: " + balance);
}
//method to deposit money
public void deposit() {
    long amt;
    System.out.println("Enter the amount you want to deposit: ");
    amt = sc.nextLong();
    balance = balance + amt;
}
//method to withdraw money
public void withdrawal() {
    long amt;
    System.out.println("Enter the amount you want to withdraw: ");
    amt = sc.nextLong();
    if (balance >= amt) {
        balance = balance - amt;
        System.out.println("Balance after withdrawal: " + balance);
    } else {
        System.out.println("Your balance is less than " + amt + "\tTransaction failed...!!" );
```

```
        }
    }
    //method to search an account number
    public boolean search(String ac_no) {
        if (accno.equals(ac_no)) {
            showAccount();
            return (true);
        }
        return (false);
    }
}
public class BankingApp {
    public static void main(String arg[]) {
        Scanner sc = new Scanner(System.in);
        //create initial accounts
        System.out.print("How many number of customers do you want to input? ");
        int n = sc.nextInt();
        BankDetails C[] = new BankDetails[n];
        for (int i = 0; i < C.length; i++) {
            C[i] = new BankDetails();
            C[i].openAccount();
        }
        // loop runs until number 5 is not pressed to exit
        int ch;
        do {
            System.out.println("\n ***Banking System Application***");
            System.out.println("1. Display all account details \n 2. Search by Account number\n
3. Deposit the amount \n 4. Withdraw the amount \n 5.Exit ");
            System.out.println("Enter your choice: ");
            ch = sc.nextInt();
                switch (ch) {
```

```
case 1:
    for (int i = 0; i < C.length; i++) {
        C[i].showAccount();
    }
    break;
case 2:
    System.out.print("Enter account no. you want to search: ");
    String ac_no = sc.next();
    boolean found = false;
    for (int i = 0; i < C.length; i++) {
        found = C[i].search(ac_no);
        if (found) {
            break;
        }
    }
    if (!found) {
        System.out.println("Search failed! Account doesn't exist..!!");
    }
    break;
case 3:
    System.out.print("Enter Account no. : ");
    ac_no = sc.next();
    found = false;
    for (int i = 0; i < C.length; i++) {
        found = C[i].search(ac_no);
        if (found) {
            C[i].deposit();
            break;
        }
    }
```

```
                if (!found) {
                    System.out.println("Search failed! Account doesn't exist..!!");
                }
                break;
            case 4:
                System.out.print("Enter Account No : ");
                ac_no = sc.next();
                found = false;
                for (int i = 0; i < C.length; i++) {
                    found = C[i].search(ac_no);
                    if (found) {
                        C[i].withdrawal();
                        break;
                    }
                }
                if (!found) {
                    System.out.println("Search failed! Account doesn't exist..!!");
                }
                break;
            case 5:
                System.out.println("See you soon...");
                break;
        }
    }
    while (ch != 5);
  }
}
```

2.

Slip No 8

1.

```
public class shpere{

    public static void main(String args[])
    {

    int radius=37;
   double pie=3.14285714286;
   double area_sphere=4*pie*(radius*radius);
        System.out.println("Surface area of sphere="+area_sphere);
     }
}
```

2. Design a screen to handle the Mouse Events such as MOUSE_MOVED and MOUSE_CLICKED and display the position of the Mouse_Click in a TextField.

```
import java.awt.*;
import java.awt.event.*;
class MyFrame extends Frame
{
        TextField t,t1;
        Label l,l1;
        int x,y;
        Panel p;
        MyFrame(String title)
        {
```

```
        super(title);
        setLayout(new FlowLayout());

        p=new Panel();
        p.setLayout(new GridLayout(2,2,5,5));
        t=new TextField(20);
        l= new Label("Co-ordinates of clicking");
        l1= new Label("Co-ordinates of movement");
        t1=new TextField(20);
        p.add(l);
        p.add(t);
        p.add(l1);
        p.add(t1);
        add(p);
        addMouseListener(new MyClick());
        addMouseMotionListener(new MyMove());
        setSize(500,500);
        setVisible(true);
}
class MyClick extends MouseAdapter
{
        public void mouseClicked(MouseEvent me)
        {
                x=me.getX();
                y=me.getY();
                t.setText("X="+x+" Y="+y);
        }
}
class MyMove extends MouseMotionAdapter
{
```

```
                public void mouseMoved(MouseEvent me)
                {
                        x=me.getX();
                        y=me.getY();
                        t1.setText("X="+ x +" Y="+y);
                }
        }
}
class Slip4
{
        public static void main(String args[])
        {
                MyFrame f = new MyFrame("Slip Number 4");
        }
}
```

Slip No 9

1. Write a program to accept a file name from command prompt, if the file exits then display number of words and lines in that file.

```
 import java.io.BufferedReader;
import java.io.FileReader;

public class CountWordFile
{
   public static void main(String[] args) throws Exception {
      String line;
      int count = 0;

      //Opens a file in read mode
```

```
        FileReader file = new FileReader("data.txt ");
        BufferedReader br = new BufferedReader(file);

        //Gets each line till end of file is reached
        while((line = br.readLine()) != null) {
            //Splits each line into words
            String words[] = line.split("");
            //Counts each word
            count = count + words.length;

        }

        System.out.println("Number of words present in given file: " + count);
        br.close();
    }
}
```

2. Write a program to display the system date and time in various formats shown below:

Current date is : 31/08/2021
Current date is : 08-31-2021
Current date is : Tuesday August 31 2021
Current date and time is : Fri August 31 15:25:59 IST 2021
Current date and time is : 31/08/21 15:25:59 PM +0530

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
public class NewClass {
public static void main(String[] args) {
    Date date = new Date();
    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
```

```
String strDate = formatter.format(date);
System.out.println("Current date is: "+strDate);

formatter = new SimpleDateFormat("MM-dd-yyyy");
strDate = formatter.format(date);
System.out.println("Current date is: "+strDate);

formatter = new SimpleDateFormat("EEEEEE MMMM dd yyyy");
strDate = formatter.format(date);
System.out.println("Current date is: "+strDate);

formatter = new SimpleDateFormat("E MMMM dd HH:mm:ss z yyyy");
strDate = formatter.format(date);
System.out.println("Current date and time is: "+strDate);

formatter = new SimpleDateFormat("dd/MM/yy HH:mm:ss a Z");
strDate = formatter.format(date);
System.out.println("Current date and time is: "+strDate);

formatter = new SimpleDateFormat("hh:mm:ss");
strDate = formatter.format(date);
System.out.println("Current time is: "+strDate);

formatter = new SimpleDateFormat("w");
strDate = formatter.format(date);
System.out.println("Current week of year is: "+strDate);

formatter = new SimpleDateFormat("W");
strDate = formatter.format(date);
System.out.println("Current week of the month is: "+strDate);
```

```
    formatter = new SimpleDateFormat("D");
    strDate = formatter.format(date);
    System.out.println("Current day of the year: "+strDate);
}
}
```

Output:

Current date is: 29/06/2018

Current date is: 06-29-2018

Current date is: Friday June 29 2018

Current date and time is: Fri June 29 22:19:13 IST 2018

Current date and time is: 29/06/18 22:19:13 PM +0530

Current time is: 10:19:13

Current week of year is: 26

Current week of the month is: 5

Current day of the year: 180

Slip No 10

1. Write a program to accept a number from the user, if number is zero then throw user definedexception “Number is 0” otherwise check whether no is prime or not (Use static keyword).

```
import java.io.*;
class NumberZeroException extends Exception
{
    public String toString()
    {
        return("Number is 0");
    }
}
```

```
class PrimeNumber
{
   int a;
   BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
   PrimeNumber()
   {
      try
      {
         System.out.println("Enter any integer to check prime ");
         a=Integer.parseInt(br.readLine());
         if(a==0)
            throw new NumberZeroException();
      }
      catch(NumberZeroException ex)
      {
         System.out.println(ex);
      }
      catch(IOException ex1)
      {
         System.out.println("Enter proper number");
      }
   }

   public void prime()
   {
      int cnt=0;
      for(int i=2;i<=a/2;i++)
         if(a%i==0)
         {
```

```
            cnt++;
            break;
          }
       if(cnt==0)
          System.out.println(a+" Number is prime");
       else
          System.out.println(a+" Number is not prime");
    }
    public static void main(String args[])
    {
       PrimeNumber pn=new PrimeNumber();
       pn.prime();
    }
}
```

```
D:\Rushi>javac PrimeNumber.java
D:\Rushi>java PrimeNumber
Enter any integer to check prime
7
7 Number is prime
```

2. Write a Java program to create a Package “SY” which has a class SYMarks (members – ComputerTotal, MathsTotal, and ElectronicsTotal). Create another package TY which has a class TYMarks (members – Theory, Practicals). Create ‘n’ objects of Student class (having rollNumber, name, SYMarks and TYMarks). Add the marks of SY and TY computer subjects and calculate the Grade (‘A’ for >= 70, ‘B’ for >= 60 ‘C’ for >= 50, Pass Class for > =40 else‘FAIL’) and display the result of the student in proper format.

```
package Assignment2.SY;

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

```
public class SYClass {
 public int ct,mt,et;
 public void get() throws IOException{
 System.out.println("Enter marks of students for computer, maths and electronics subject out of 200 ");
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
 ct=Integer.parseInt(br.readLine());
 mt=Integer.parseInt(br.readLine());
 et=Integer.parseInt(br.readLine());
 }
  }

package Assignment2.TY;

import java.io.*;
public class TYClass {
 public int tm,pm;
 public void get() throws IOException{
 System.out.println("Enter the marks of the theory out of 400 and practicals out of 200: ");
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
 tm=Integer.parseInt(br.readLine());
 pm=Integer.parseInt(br.readLine());
 }
 }

package Assignment2;
import Assignment2.SY.*;
import Assignment2.TY.*;
import java.io.*;
class StudentInfo{
```

```
 int rollno;
 String name,grade;
 public float gt,tyt,syt;
 public float per;
 public void get() throws IOException{
 System.out.println("Enter roll number and name of the student: ");
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
 rollno=Integer.parseInt(br.readLine());
 name=br.readLine();
 }

}
public class StudentMarks {
 public static void main(String[] args) throws IOException{

 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
 System.out.println("Enter the number of students:");
 int n=Integer.parseInt(br.readLine());
 SYClass sy[]=new SYClass[n];
 TYClass ty[]=new TYClass[n];
 StudentInfo si[]=new StudentInfo[n];
 for(int i=0;i<n;i++)
 {
 si[i]=new StudentInfo();
 sy[i]=new SYClass();
 ty[i]=new TYClass();

 si[i].get();
 sy[i].get();
 ty[i].get();
```

```
 si[i].syt=sy[i].ct+sy[i].et+sy[i].mt;
 si[i].tyt=ty[i].pm+ty[i].tm;
 si[i].gt=si[i].syt+si[i].tyt;
 si[i].per=(si[i].gt/1200)*100;

 if(si[i].per>=70) si[i].grade="A";
 else if(si[i].per>=60) si[i].grade="B";
 else if(si[i].per>=50) si[i].grade="C";
 else if(si[i].per>=40) si[i].grade="Pass";
 else si[i].grade="Fail";
   }
 System.out.println("Roll No\tName\tSyTotal\tTyTotal\tGrandTotal\tPercentage\tGrade");
 for(int i=0;i<n;i++)
 {
System.out.println(si[i].rollno+"\t"+si[i].name+"\t"+si[i].syt+"\t"+si[i].tyt+"\t"+si[i].gt+"\t\t"
+si[i].per+"\t\t"+si[i].grade);
 }
 }
}
```

Slip No 11

1. Write a program to implement Border Layout Manager.

```
import java.awt.*;
import javax.swing.*;

public class Border
{
JFrame f;
Border()
{
   f = new JFrame();
```

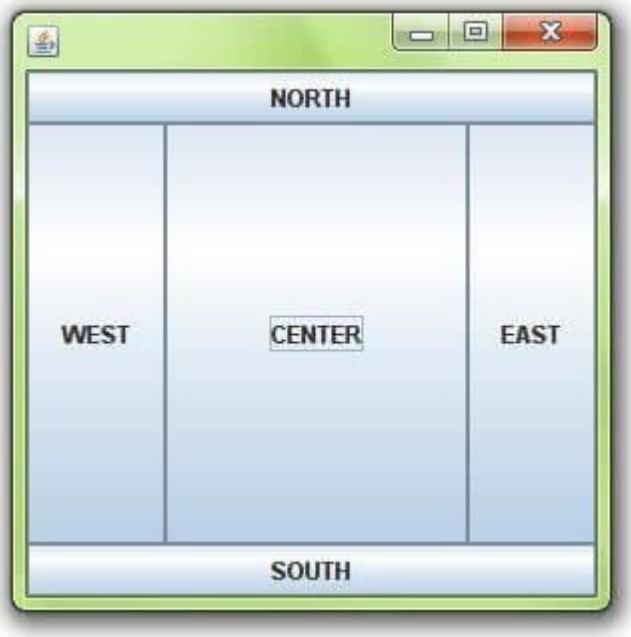
```
    // creating buttons
    JButton b1 = new JButton("NORTH");; // the button will be labeled as NORTH
    JButton b2 = new JButton("SOUTH");; // the button will be labeled as SOUTH
    JButton b3 = new JButton("EAST");; // the button will be labeled as EAST
    JButton b4 = new JButton("WEST");; // the button will be labeled as WEST
    JButton b5 = new JButton("CENTER");; // the button will be labeled as CENTER

    f.add(b1, BorderLayout.NORTH); // b1 will be placed in the North Direction
    f.add(b2, BorderLayout.SOUTH);  // b2 will be placed in the South Direction
    f.add(b3, BorderLayout.EAST);  // b2 will be placed in the East Direction
    f.add(b4, BorderLayout.WEST);  // b2 will be placed in the West Direction
    f.add(b5, BorderLayout.CENTER);  // b2 will be placed in the Center

    f.setSize(300, 300);
    f.setVisible(true);
}
public static void main(String[] args) {
    new Border();
}
}
```

Output:

Output:



2. Define a class CricketPlayer (name,no_of_innings,no_of_times_notout, totatruns,

bat_avg).Create an array of n player objects. Calculate the batting average for each player using staticmethod avg(). Define a static sort method which sorts the array on the basis of average.Display the player details in sorted order.

```
import java.io.*;
class Cricket {
String name;
int inning, tofnotout, totalruns;
float batavg;
public Cricket(){
name=null;
inning=0;
tofnotout=0;
totalruns=0;
batavg=0;
}
public void get() throws IOException{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter the name, no of innings, no of times not out, total runs: ");
name=br.readLine();
inning=Integer.parseInt(br.readLine());
tofnotout=Integer.parseInt(br.readLine());
totalruns=Integer.parseInt(br.readLine());
}
public void put(){
System.out.println("Name="+name);
System.out.println("no of innings="+inning);
System.out.println("no times notout="+tofnotout);
System.out.println("total runs="+totalruns);
System.out.println("bat avg="+batavg);
```

```
}
static void avg(int n, Cricket c[]){
try{
for(int i=0;i<n;i++){
c[i].batavg=c[i].totalruns/c[i].inning;
}
}catch(ArithmeticException e){
System.out.println("Invalid arg");

}
}
static void sort(int n, Cricket c[]){
String temp1;
int temp2,temp3,temp4;
float temp5;
for(int i=0;i<n;i++){
for(int j=i+1;j<n;j++){
if(c[i].batavg<c[j].batavg){
temp1=c[i].name;
c[i].name=c[j].name;
c[j].name=temp1;
temp2=c[i].inning;
c[i].inning=c[j].inning;
c[j].inning=temp2;
temp3=c[i].tofnotout;
c[i].tofnotout=c[j].tofnotout;
c[j].tofnotout=temp3;
temp4=c[i].totalruns;
c[i].totalruns=c[j].totalruns;
c[j].totalruns=temp4;
```

```
temp5=c[i].batavg;
c[i].batavg=c[j].batavg;
c[j].batavg=temp5;
}
}
}
}
}
public class a4sa1 {
public static void main(String args[])throws IOException{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter the limit:");
int n=Integer.parseInt(br.readLine());
Cricket c[]=new Cricket[n];
for(int i=0;i<n;i++){
c[i]=new Cricket();
c[i].get();
}
Cricket.avg(n,c);
Cricket.sort(n, c);
for(int i=0;i<n;i++){
c[i].put();
}
}
}
```

Output:

```
Enter the limit:
2
Enter the name, no of innings, no of times not out, total runs:
```

Virat

3

1

100

Enter the name, no of innings, no of times not out, total runs:

Rohit

2

1

120

Name=Rohit

no of innings=2

no times notout=1

total runs=120

bat avg=60.0

Name=Virat

no of innings=3

no times notout=1

total runs=100

bat avg=33.0

Slip No 12

1. Define a class MyDate(Day, Month, year) with methods to accept and display a MyDateobject.Accept date as dd,mm,yyyy. Throw user defined exception "InvalidDateException" if the date is invalid.

```
import java .io.*;
class InvalidDateException extends Exception
{
}
class MyDate
{
   int day,mon,yr;
```

```
    void accept(int d,int m,int y)
    {
       day=d;
       mon=m;
       yr=y;
    }
    void display()
    {
       System.out.println("Date is valid : "+day+"/"+mon+"/"+yr);
    }
}
class DateMain
{
   public static void main(String arg[]) throws Exception
   {
      System.out.println("Enter Date  :  dd mm yyyy ");
      BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
      int day=Integer.parseInt(br.readLine());
      int mon=Integer.parseInt(br.readLine());
      int yr=Integer.parseInt(br.readLine());
      int flag=0;
      try
         {
               if(mon<=0 || mon>12)

                  throw  new InvalidDateException();
               else
               {
                  if(mon==1 || mon==3 || mon==5 || mon==7 || mon==8 || mon==10 ||
mon==12)
                  {
                     if(day>=1 && day <=31)
                        flag=1;
                     else
                        throw  new InvalidDateException();
                  }
                  else if (mon==2)
                  {
```

```
                if(yr%4==0)
                {
                   if(day>=1 && day<=29)
                      flag=1;
                   else throw  new InvalidDateException();
                }
                else
                {
                   if(day>=1 && day<=28)
                      flag=1;
                   else throw  new InvalidDateException();
                }
             }
             else
             {
                if(mon==4 || mon == 6 || mon== 9 || mon==11)
                {
                   if(day>=1 && day <=30)
                      flag=1;
                   else throw  new InvalidDateException();
                }
             }
          }
        if(flag== 1)
        {
           MyDate dt = new MyDate();
           dt.accept(day,mon,yr);
           dt.display();
        }
     }
   catch (InvalidDateException mm)
   {
      System.out.println("Invalid Date");
   }

}
```

2. Create an employee class(id,name,deptname,salary). Define a default and parameterized constructor. Use ‘this’ keyword to initialize instance variables. Keep a count of objects created. Create objects using parameterized constructor and display the object count aftereach object is created. (Use static member and method). Also display the contents of each object.

```
import java.util.Scanner;
public class Employee
{
int id;
String name;
String deptname;
float salary;
static int numberofobjects=0;
Employee()
{
id=0;
name="";
deptname="";
salary=0;
}
Employee(int id,String name,String deptname,float salary )
{
this.id=id;
this.name=name;
this.deptname=deptname;
this.salary=salary;
numberofobjects++;
}
public void display()
{
System.out.println("Employee Id :"+id);
System.out.println("Employee name: "+name);
System.out.println("Employee Department: "+deptname);
System.out.println("Employee Salary :"+salary);
}
public static void main(String[] args){
int n=0;
Scanner sc=new Scanner(System.in);
System.out.print("How many employees you want to enter :");
n=sc.nextInt();
Employee[] ob=new Employee[n];
for(int i=0;i<n;i++)
{
sc= new Scanner(System.in);
System.out.println("Enter Id of employee "+(i+1)+" :");
int id=sc.nextInt();
System.out.println("Enter Name of employee "+(i+1)+" :");
sc.nextLine();
String name= sc.nextLine();
```

```
System.out.println("Enter dept name of employee "+(i+1)+" :");
String deptname=sc.nextLine();
System.out.println("Enter salary of employee "+(i+1)+" :");
float salary = sc.nextFloat();
ob[i]=new Employee(id,name,deptname,salary);
System.out.println("\nNumber of Objects : "+numberofobjects);

}
for(int i=0;i<n;i++)
{
ob[i].display();
}
}
}
```

Output:

```
How many employees you want to enter :2
Enter Id of employee 1 :
1
Enter Name of employee 1 :
rohit
Enter dept name of employee 1 :
abc
Enter salary of employee 1 :
1000000
Number of Objects : 1
Enter Id of employee 2 :
2
Enter Name of employee 2 :
bairwa
Enter dept name of employee 2 :
90000000
Enter salary of employee 2 :
78888888
Number of Objects : 2
Employee Id :1
Employee name: rohit
Employee Department: abc
Employee Salary :1000000.0
Employee Id :2
Employee name: bairwa
Employee Department: 90000000
Employee Salary :7.8888888E7
```

Slip No 13

1. Accept the names of two files and copy the contents of the first to the second. First file having Book name and Author name in file.

```
import java.io.*;
import java.util.*;
class copy{
public static void main(String arg[]) throws Exception {
Scanner sc = new Scanner(System.in);
System.out.print("source file name :\n");
String file1 = sc.next();
System.out.print("destination file name :\n");
String file2 = sc.next();
FileReader fin = new FileReader(file1);
FileWriter fout = new FileWriter(file2, true);
int c;
while ((c = fin.read()) != -1) {
fout.write(c);
}
System.out.println("Copy finish...");

fin.close();
fout.close();
}
}
```

Output:

```
source file name :
bookinfo.txt
destination file name :
info.txt
```

Copy finish...

2. Write a program to define a class Account having members custname, accno. Define default and parameterized constructor. Create a subclass called SavingAccount with member savingbal,minbal. Create a derived class AccountDetail that extends the class SavingAccount with members, depositamt and withdrawalamt. Write a appropriate method to display customer details.