WAJP to create a class ‘Student’, overload the constructors depending on the student is final year or not, if yes take a parameter companyName, else skip this parameter. Also, overload a method Eligible(), taking attendance,other overloaded method having attendance and Internal marks percentage as parameters. Also define amethod calcCGPA() which takes Student object as a parameter/’this’, also sendDetails() that returns the student object. Use ‘this’ keyword to handle the instance variable hiding and constructor chaining/reusing.

class Student

{

String sname,cname; // **instance/object level variables**

int rno;

int a[];

Static String collgName = “VCE”; // **static/class level variables**

Student(String sname,int rollno)

{

**this.sname = sname; //instance variable hiding**

rno = rollno;

}

**Student( String sname,int rno,String cname) // constructor overloading**

{

**this(sname, rno); // constructor chaining**

this.cname = cname;

}

/\*int[] readCalc(int totSub) // arrays

{

a = new int[totSub];

**//** after intiliazing array **use recursion to print those array elements**

} \*/

**void eligibleCheck(int att,int intPer) // local variables**

{

if(att<=75 && att<=65&& intPer<40)

System.out.println("Eligible..But submit medical certificate");

else if(att>=75 && intPer>=40)

System.out.println("Eligible.");

else

System.out.println("Sorry..Detained");

}

**void eligibleCheck(int mperc, int blg, boolean placed) //method overloading**

{

if( mperc > 75 && blg==0 && placed==false)

System.out.println("Eligible.. to attend this placement");

else

System.out.println("Not Eligible.. cannot attend");

}

Student verify()

{

**return this;** // returning object of current class using 'this'

}

}

public class Main

{

static void display(**Student obj)** // passing object as a parameter

{

System.out.println("Name="+obj.sname);

System.out.println("Roll Number="+obj.rno);

}

public static void main(String[] args) {

Student s1=new Student("Priya",2498,"Google");

Student s2=new Student("Testname",1234);

System.out.println("Data of college: “+ **Student.collgName); // accessing static variable**

System.out.println("Checking eligibility.. of"+**s1.sname); // accessing instance variable**

s1.eligibleCheck(84,69);

s1.eligibleCheck(85,0,true);

Student s3 = s1.verify();

display(s3);

System.out.println("Checking eligibility.. of"+s2.sname);

s1.eligibleCheck(40,69);

s1.eligibleCheck(75,0,false);

Student s4 = **s2.verify();**

display(s4);

s4 = null;

System.gc**(); //garbage collection**

}

}

Recursion

class Rec

{

int a[];

Rec(int n)

{

a = new int[n];

}

void print(int i)

{

if( i == 0)

return;

else

print(i-1);

System.out.println(a[i-1]);

}

public static void main(String[] arg)

{

Rec obj = new Rec(5);

for(int x=0;x<obj.a.length;x++)

obj.a[x] = x\*3;

obj.print(obj.a.length);

}

} /\* 0 3 6 9 12 \*/

**INHERITANCE**

|  |  |
| --- | --- |
| **Class Student**  **{**  **String sname;**  **Int rollNo;**  **Boolean scholarship;**  **Void display() { …}**  **}**  **Class PG extends Student**  **{**  **Boolean GateQ, PGEQ, scholarship;**  **PG(String sn, Int rno, String specilaization)**  **{**  **Super(sn,rno);**  **}**  **//OVERRIDING**  **Void display() { super.display()…**  **Super. scholarship; }**  **}**  **Class UG extends Student**  **{**  **String branch;**  **Boolean companyPlaced;**  **}** | **Class Account**  **{**  **Void rateOfInterest() { … }**  **}**  **Class HomeLoanAcc extends Account**  **{**  **Void rateOfInterest() { … }**  **}**  **Class PersonalLoanAcc extends Account**  **{**  **Void rateOfInterest() { … }**  **}**  **Class Test**  **{**  **Psvm(String[] a)**  **{**  **Account a = new PersonalLoanAcc();**  **a.rateOfInterest();**  **// DYNAMIC METHOD DISPATCH** |