Part A

1st Question:

```
 import java.util.*;
abstract class Vehicle
{
   boolean hashelmet;
   int yom;
   abstract void getData();
   abstract void putData();
   Vehicle(boolean h,int n)
   {
      hashelmet=h;
      yom=n;
   }
}
class TwoWheeler extends Vehicle
{   private String Brand;
   protected int Cost;
   String EngineType;
   public String Color;
   TwoWheeler(int n)
   {
      super(true,n);
   }
   void getData()
   {   Scanner sc=new Scanner(System.in);
      System.out.println("Enter Brand name,Cost,EngineType and Colour");
      Brand=sc.next();
      Cost=sc.nextInt();
      EngineType=sc.next();
      Color=sc.next();
   }
   void putData()
   {

System.out.println("Brand:"+Brand+"\nCost:"+Cost+"\nEngineType:"+EngineType+"\nColor:"+C
olor+"\nYear of Manufacture:"+yom+"\nHas helmet:"+hashelmet);
   }
}
final class FourWheeler extends Vehicle
{
   FourWheeler(int n)
   {
```

```
        super(false,n);
    }
    void getData()
    {
    }
    void putData()
    {
        System.out.println("Year of Manufacture:"+yom+"\nHas helmet:"+hashelmet);
    }
}
class MyTwoWheeler extends TwoWheeler
{
    String name;
    MyTwoWheeler(String name,int n)
    {
        super(n);
        this.name=name;
    }
    void display()
    {
        getData();
        putData();
        System.out.println("Name:"+name);
    }
}

/*class A extends FourWheeler
{
    A()
    {
        super(5);
    }
}
*/
public class Main
{
            public static void main(String[] args) {
                    /*Vehicle v=new Vehicle();
                    Cannot Create instance of an abstract class*/
                    TwoWheeler t1=new TwoWheeler(1995);
        FourWheeler f1=new FourWheeler(2006);
        t1.getData();
        t1.putData();
```

```
            f1.putData();
        }
}
```

2nd Question:

```
abstract class Shape
{
    String color;
    boolean filled;
    abstract double getArea();
    Shape()
    {
        color="green";
        filled=true;
    }
    Shape(String c,boolean f)
    {
        color=c;
        filled=f;
    }
    boolean isFILLED()
    {
        return filled;
    }
    String getColor()
    {
        return color;
    }
    void setFILLED(boolean b)
    {
        this.filled=b;
    }
    void setColor(String c)
    {
        this.color=c;
    }
    public String toString()
    {
        if(this.filled==false)
            return "A Shape with color " +this.color+" and not filled";
        else
            return "A Shape with color " +this.color+" and filled";
    }
```

```
}
class Circle extends Shape
{
   int r;
   Circle(int r1)
   {   super();
       r=r1;
   }
   Circle(String c,boolean f,int r1)
   {   super(c,f);
       r=r1;
   }
   double getArea()
   {
       return 3.14*r*r;
   }
   void display()
   {
       System.out.println(isFILLED());
       System.out.println(getColor());
   }
   void change(String c,boolean b)
   {
       setColor(c);
       setFILLED(b);
   }
}
final class Rectangle extends Shape
{
   int a,b;
   Rectangle(int a1,int b1)
   {   super();
       a=a1;
       b=b1;
   }
   Rectangle(String c,boolean f,int a1,int b1)
   {   super(c,f);
       a=a1;
       b=b1;
   }
   double getArea()
   {
       return a*b;
```

```
   }
}
/*class Square extends Rectangle
{   Square()
   {
      super(5,6);
   }
   void display()
   {
      System.out.println(a + " " + b);
   }
}
Cannot inherit a final class
*/
public class Main
{
        public static void main(String[] args) {
                /*Shape s=new Shape();
                CAnnor create instance of an abstact class*/
                Circle c=new Circle("blue",false,5);
                Rectangle r=new Rectangle("red",true,2,4);
                System.out.println(c);
                System.out.println(r);
                System.out.println(c.getArea());
                System.out.println(r.getArea());
            c.display();
            c.change("brown",true);
            c.display();
        }
}
```

3rd Qustion:

```
import java.util.*;
abstract class Student
{
   private String Name;
   protected int ID;
   double grade;
   public int age;
   abstract boolean isPassed(double Grade);
   void setter(String name)
   {
      Name=name;
```

```
    }
    String getter()
    {
        return Name;
    }
}
class Undergrad extends Student
{
    void getData()
    {   Scanner sc=new Scanner(System.in);
        System.out.println("Enter Name,ID,age");
        setter(sc.next());
        ID=sc.nextInt();
        age=sc.nextInt();
    }
    boolean isPassed(double Grade)
    {
        grade=Grade;
        if(grade<=70)
            return false;
        else
            return true;
    }
    void display()
    {
        System.out.println("Name:"+getter()+"\nAge:"+age+"\nID:"+ID);
    }
    /*void setter(String name)
    {
    }
    Cannot override the final setter method*/
}
class Grad extends Student
{
    void getData()
    {   Scanner sc=new Scanner(System.in);
        System.out.println("Enter Name,ID,age");
        setter(sc.next());
        ID=sc.nextInt();
        age=sc.nextInt();
    }
    boolean isPassed(double Grade)
    {
```

```
        grade=Grade;
        if(grade<=70)
            return false;
        else
            return true;
    }
    void display()
    {
        System.out.println("Name:"+getter()+"\nAge:"+age+"\nID:"+ID);
    }
}
public class Main
{
            public static void main(String[] args) {
                    Undergrad u=new Undergrad();
                    u.getData();
                if(u.isPassed(65))
                    System.out.println("Student has passed");
                else
                    System.out.println("Student has failed");
                u.display();
        Grad g=new Grad();
        g.getData();
                if(g.isPassed(90))
                    System.out.println("Student has passed");
                else
                    System.out.println("Student has failed");
                g.display();
            }
}
```

4th question:

```
class Car
{
    int speed;
    double regularPrice;
    String color;
    Car(int s,double price,String c)
    {
        speed=s;
        regularPrice=price;
        color=c;
    }
```

```
    double getSalePrice()
    {
        return regularPrice;
    }
}
class Truck extends Car
{   int weight;
    Truck(int s,double price,String c,int w)
    {
        super(s,price,c);
        weight=w;
    }
    double getSalePrice()
    {
        if(weight>2000)
        {    regularPrice=regularPrice*0.9;
            return regularPrice;
        }
        else
        {    regularPrice=regularPrice*0.8;
            return regularPrice;
        }
    }
}
class Ford extends Car
{   int manufacturerDiscount;
    Ford(int s,double price,String c,int m)
    {
        super(s,price,c);
        manufacturerDiscount=m;
    }
    double getSalePrice()
    {
        regularPrice-=manufacturerDiscount;
        return regularPrice;
    }
}
class Sedan extends Car
{   int length;
    Sedan(int s,double price,String c,int l)
    {
        super(s,price,c);
        length=l;
```

```
    }
    double getSalePrice()
    {
        if(length>20)
        {    regularPrice=regularPrice*0.95;
            return regularPrice;
        }
        else
        {    regularPrice=regularPrice*0.9;
            return regularPrice;
        }
    }
}
public class Main
{
        public static void main(String[] args) {
                Truck t=new Truck(65,2500000,"Red",3000);
                System.out.println("Price of truck is "+t.getSalePrice());
                Ford f=new Ford(120,2200000,"Yellow",120000);
                System.out.println("Price of ford is "+f.getSalePrice());
                Sedan s= new Sedan(100,3500000,"Blue",22);
                System.out.println("Price of Sedan is "+s.getSalePrice());
        }
}
```

5th Question:

```
class SavingsAccount
{
    static int annualInterestRate;
    private double savingsBalance;
    SavingsAccount(double s)
    {
        savingsBalance=s;
    }
    static void modifyInterestRate(int x)
    {
        annualInterestRate=x;
    }
    void calculateMonthlyInterest()
    {
        double d=(savingsBalance*annualInterestRate)/12;
        savingsBalance=savingsBalance+d;
    }
```

```
    void display()
    {
        System.out.println(savingsBalance);
    }
}
public class Main
{
        public static void main(String[] args) {
                SavingsAccount saver1=new SavingsAccount(2000);
                SavingsAccount saver2=new SavingsAccount(3000);
                SavingsAccount.modifyInterestRate(4);
                saver1.calculateMonthlyInterest();
                saver2.calculateMonthlyInterest();
                saver1.display();
                saver2.display();
                SavingsAccount.modifyInterestRate(5);
                saver1.calculateMonthlyInterest();
                saver2.calculateMonthlyInterest();
                saver1.display();
                saver2.display();
        }
}
```

6th Question:

```
class Customer
{
    private int ID;
    private String Name;
    private int discount;
    Customer(int ID,String Name,int discount)
    {
        this.Name=Name;
        this.ID=ID;
        this.discount=discount;
    }
    int getID()
    {
        return ID;
    }
    String getName()
    {
        return Name;
```

```
    }
    int getDiscount()
    {
        return discount;
    }
    void setDiscount(int discount)
    {
        this.discount=discount;
    }
    public String toString()
    {
        return Name+"("+ID+")";
    }
}
class Invoice
{
    private int ID;
    private Customer customer;
    private double amount;
    Invoice(int ID,Customer customer,double amount)
    {
        this.ID=ID;
        this.customer=customer;
        this.amount=amount;
    }
    int getID()
    {
        return ID;
    }
    Customer getCustomer()
    {
        return customer;
    }
    void setCustomer(Customer customer)
    {
        this.customer=customer;
    }
    String getAmount()
    {
        return Double.toString(amount);
    }
    void setAmount(double amount)
    {
```

```
        this.amount=amount;
    }
    String getCustomerName()
    {
        return customer.getName();
    }
    double getAmountAfterDiscount()
    {
        return (amount*customer.getDiscount())/100;
    }
}
public class Main
{
            public static void main(String[] args) {
                Customer c=new Customer(25,"AAAA",5);
                System.out.println(c.getID());
                System.out.println(c.getDiscount());
                System.out.println(c.getName());
                c.setDiscount(7);
                System.out.println(c.getDiscount());
                System.out.println(c);
                Customer c1=new Customer(26,"BBBBBB",9);
                Invoice i=new Invoice(28,c1,60000);
                System.out.println(i.getID());
                System.out.println(i.getCustomer());
                System.out.println(i.getAmount());
                i.setAmount(70000);
                System.out.println(i.getAmount());
                System.out.println(i.getCustomerName());
                System.out.println(i.getAmountAfterDiscount());
            }
}
```

7th Question:

```
class Person
{
  private String name;
  private String address;
    Person (String name, String address)
  {
    this.name = name;
    this.address = address;
  }
```

```
  String getName ()
  {
    return name;
  }
  String getAddress ()
  {
    return address;
  }
  void setAddress (String address)
  {
    this.address = address;
  }
  public String toString ()
  {
    return name + "(" + address + ")";
  }
}

class Student extends Person
{
  int numCourses = 0;
  String courses[] = new String[30];
  int grades[] = new int[30];
    Student (String name, String address)
  {
    super (name, address);
  }
  void addCourseGrade (String course, int grade)
  {
    if (numCourses <= 29)
      {
          courses[numCourses] = course;
          grades[numCourses] = grade;
          numCourses++;
      }
    else
      {
          System.out.println ("Maximum number of courses");
      }
  }
  void printGrades ()
  {
    for (int i = 0; i < numCourses; i++)
```

```
    System.out.println ("Course:" + courses[i] + " Grade:" + grades[i]);
  }
  double getAverageGrades ()
  {
    double d = 0;
    for (int i = 0; i < numCourses; i++)
      d = d + grades[i];
    d = d / numCourses;
    return d;
  }
  public String toString ()
  {
    return getName () + "(" + getAddress () + ")";
  }
}

class Teacher extends Person
{
  int numCourses = 0;
  String courses[] = new String[5];
    Teacher(String name, String address)
  {
    super (name, address);
  }
  boolean addCourse (String course)
  {
    if (numCourses <= 4)
      {
          for (int i = 0; i < numCourses; i++)
            if (courses[i].equals (course))
              return false;
          courses[numCourses] = course;
          numCourses++;
          return true;
      }
    else
      return false;
  }
  boolean removeCourse (String course)
  {
    if (numCourses != 0)
       {
                for (int i = 0; i < numCourses; i++)
```

```
                    if (courses[i].equals (course))
                    {
                      courses[i] = " ";
                      return true;
                    }
                return false;
      }
    return false;
  }
public String toString ()
{
              return getName () + "(" + getAddress () + ")";}
}
public class Main
{
           public static void main (String[]args)
               {
                 Student s = new Student ("AA", "BB");
                 System.out.println (s);
                 s.addCourseGrade ("Maths", 85);
                 s.addCourseGrade ("OOPS", 80);
          s.addCourseGrade ("DS",75);
          s.addCourseGrade ("DMS", 70);
          s.printGrades();
          System.out.println(s.getAverageGrades());
          Teacher t=new Teacher("CC","DD");
          if(t.addCourse("Maths"))
             System.out.println("Course added");
          else
             System.out.println("Max limit reached/course already exists");
          if(t.addCourse("Maths"))
             System.out.println("Course added");
          else
             System.out.println("Max limit reached/course already exists");
          if(t.addCourse("OOPS"))
             System.out.println("Course added");
          else
             System.out.println("Max limit reached/course already exists");
          if(t.addCourse("DS"))
             System.out.println("Course added");
          else
             System.out.println("Max limit reached/course already exists");
          if(t.removeCourse("Maths"))
```

```
            System.out.println("Course removed");
        else
            System.out.println("Zero courses/course does not exist");
               if(t.removeCourse("TOC"))
            System.out.println("Course removed");
        else
            System.out.println("Zero courses/course does not exist");
             }
}
```

8th Question

```
import java.util.*;
class Record
{   static int n;
    public String name[];
    public int rank[];
    Record()
    {
    }
    Record(int num_of_records)
    {
        n=num_of_records;
        name=new String[num_of_records];
        rank=new int[num_of_records];
    }
    void readvalues(Scanner sc)
    {   for(int i=0;i<n;i++)
        {
            System.out.println("Enter name and marks of record "+ (i+1));
            name[i]=sc.next();
            rank[i]=sc.nextInt();
        }
    }
    void display()
    {
        for(int i=0;i<n;i++)
            System.out.println("Name:"+name[i]+" Rank:"+rank[i]);
    }
}
class Rank extends Record
{
    int index;
    Rank(int num_of_records)
```

```
    {   super(num_of_records);
        index=0;
    }
    void highest()
    {   int topr=rank[0];
        for(int i=1;i<n;i++)
        {
            if(rank[i]>topr)
            {
                topr=rank[i];
                index=i;
            }
        }
    }
    public String toString()
    {
        return name[index]+","+rank[index];
    }
}
public class Main
{
            public static void main(String[] args) {
                Scanner sc=new Scanner(System.in);
                Rank ra=new Rank(5);
                ra.readvalues(sc);
                ra.display();
                ra.highest();
                System.out.println(ra);
            }
}
```

9th Question:

```
abstract class Reservation
{
    abstract boolean reserve(int s,String type);
    abstract int getAvailableSeats(String s);
}
class ReserveBus extends Reservation
{
    int totalSeats;
    int n=0;
    ReserveBus(int totalSeats)
    {
```

```
        this.totalSeats=totalSeats;
    }
    boolean reserve(int s,String type)
    {
        if(s<getAvailableSeats(type))
        {
            n=n+s;
            return true;
        }
        else
        {
            return false;
        }
    }
    int getAvailableSeats(String s)
    {
        return totalSeats-n;
    }
}
class ReserveTrain extends Reservation
{
    int upperBerthTotalSeats,middleBerthTotalSeats,lowerBerthTotalSeats;
    int nu=0,nm=0,nl=0;
    ReserveTrain(int upperBerthTotalSeats,int middleBerthTotalSeats,int lowerBerthTotalSeats)
    {
        this.upperBerthTotalSeats=upperBerthTotalSeats;
        this.middleBerthTotalSeats=middleBerthTotalSeats;
        this.lowerBerthTotalSeats=lowerBerthTotalSeats;
    }
    boolean reserve(int s,String type)
    {
        switch (type){
              case "Upper":{
                        if(s<getAvailableSeats(type))
                        {
                            nu=nu+s;
                            return true;
                        }
                        else
                        {
                        return false;
                        }
                       }
```

```
            case "Middle":{
                        if(s<getAvailableSeats(type))
                        {
                            nm=nm+s;
                            return true;
                        }
                        else
                        {
                        return false;
                        }
                       }
            case "Lower": {
                        if(s<getAvailableSeats(type))
                        {
                            nl=nl+s;
                            return true;
                        }
                        else
                        {
                        return false;
                        }
                       }
            default:return false;
        }
    }
    int getAvailableSeats(String s)
    {
        switch (s){
            case "Upper":{
                        return upperBerthTotalSeats-nu;
                    }
            case "Middle":{
                        return middleBerthTotalSeats-nm;
                    }
            case "Lower":{
                        return lowerBerthTotalSeats-nl;
                    }
            default:return -1;
        }
    }
}
public class Main
{
```

```
	public static void main(String[] args) {
		ReserveBus rb=new ReserveBus(25);
		if(rb.reserve(12,"sitting"))
		   System.out.println("Booked");
		else
		   System.out.println("Not booked");
		System.out.println(rb.getAvailableSeats("Sitting"));
		ReserveTrain rt=new ReserveTrain(25,25,25);
		if(rt.reserve(12,"Upper"))
		   System.out.println("Booked");
		else
		   System.out.println("Not booked");
		System.out.println(rt.getAvailableSeats("Upper"));
		if(rt.reserve(12,"Middle"))
		   System.out.println("Booked");
		else
		   System.out.println("Not booked");
		System.out.println(rt.getAvailableSeats("Middle"));
		if(rt.reserve(12,"Lower"))
		   System.out.println("Booked");
		else
		   System.out.println("Not booked");
		System.out.println(rt.getAvailableSeats("Lower"));
	}
}
```

10th Question:

```
class Faculty
{
	public String name;
	private int basic;
	public double salary;
	public Faculty(int basic,String name)
	{
		this.basic=basic;
		this.name=name;
	}
	public String getDetails()
	{
		return(name+" "+getSalary());
	}
	public double getSalary()
```

```
	{
		salary=basic;
		return(salary);
	}
}

class AssistantProfessor extends Faculty
{
	public int DA;
	public AssistantProfessor(int da,int b,String s)
	{
		super(b,s);
		DA=da;
	}
	public double getSalary()
	{
		return(super.getSalary()+((super.getSalary()*DA)/100));
	}
	public String getDetails()
	{
		return(name+" "+getSalary());
	}
}

class AssociateProfessor extends AssistantProfessor
{
	public int MedAllowance;
	public AssociateProfessor(int ma,int da,int b,String s)
	{				super(da,b,s);
			MedAllowance=ma;
	}
	public double getSalary()
	{
		return(super.getSalary()+MedAllowance);
	}
	public String getDetails()
	{
		return(name+" "+getSalary());
	}
}

class Professor extends AssociateProfessor
{
```

```
	public int OtherAllowance;
	public Professor(int oa,int ma,int da,int b,String s)
	{
		super(ma,da,b,s);
		OtherAllowance=oa;
	}
	public double getSalary()
	{
		return(super.getSalary()+(OtherAllowance*0.01*super.getSalary()));
	}
	public String getDetails()
	{
		return(name+" "+getSalary());
	}
}
public class Main
{
	public static void main(String[] args)
	{
		Professor p=new Professor(5,3,2,60000,"AAAA");
		System.out.println(p.getSalary());
		System.out.println(p.getDetails());
	}
}
```

11th Question:

```
package p1;
public class Student
{
   public String USN="1MS18CS300";
   public String DepartmentName="CSE";
   public int su1=35;
   public int su2=36;
   public int su3=37;
   double SGPA=8.5;
   public void display()
   {
      System.out.println("USN:"+USN);
      System.out.println("Subject 1 Marks:"+su1);
      System.out.println("Subject 2 Marks:"+su2);
```

```
        System.out.println("Subject 3 Marks:"+su2);
        System.out.println("Department:"+DepartmentName);
        System.out.println("SGPA:"+SGPA);
    }
}
```

```
package p1;
public class Teacher
{
    public String StaffID="7000";
    public String StaffName="AAAa";
    public String Designation="Professor";
    public String Subject[]={"Maths","Physcis"};
    public void display()
    {
        System.out.println("Staff ID:"+StaffID);
        System.out.println("StaffName:"+StaffName);
        System.out.println("Designation:"+Designation);
        System.out.println("Subject 1:"+Subject[1]);
        System.out.println("Subject 2:"+Subject[2]);
    }
}
```

```
import p1.Student;
import p1.Teacher;
public class Main {
        public static void main(String[] args) {
                Student s=new Student();
                System.out.println(s.USN);
                s.display();
                Teacher t=new Teacher();
                System.out.println(t.Subject[1]);
                t.display();
        }
}
```

12th Question:

```
package AdvMath;
import java.lang.Math;
```

```
public class Sum {
	public void sum(int x)
	{
		System.out.println(Math.sin(x)+Math.cos(x)+Math.tan(x));
	}
}
```

```
package AdvMath;

public class Triples {
	public void display(int limit)
   {
		int a, b, c = 0;
		int m = 2;
		while (c < limit) {
			for (int n = 1; n < m; ++n) {
				a = m * m - n * n;
				b = 2 * m * n;
				c = m * m + n * n;
				if (c > limit)
					break;
				System.out.println(a + " " + b + " " + c);
			}
	m++;
      }
   }

}
```

```
import AdvMath.Sum;
import AdvMath.Triples;;
public class Main {
	public static void main(String[] args) {
		Sum s=new Sum();
		s.sum(0);
		Triples t=new Triples();
		t.display(20);
	}
}
```