Name : Gokul Sarkar
Roll No : 46

Assignment - 5

1. Design an abstract class having two methods. Create Rectangle and Triangle classes by inheriting the shape class and override the above methods to suitably implement for Rectangle and Triangle class.

```
abstract class shape {
   abstract double area();
   abstract double perimeter();
}
class Rectangle extends shape{
   double length, width;
   Rectangle(double length, double width ){
      this.length = length;
      this.width = width;
   }
   @Override
   double area(){
      return length * width;
   }

   @Override
   double perimeter(){
      return 2*(length + width);
   }
}
class Triangle extends shape{
   double side1, side2, side3, height;
   Triangle(double side1, double side2, double side3, double height){
      this.side1 = side1;
      this.side2 = side2;
      this.side3 = side3;
      this.height = height;
   }
   @Override
   double area(){
```

Name : Gokul Sarkar
Roll No : 46

```
        return side1 + side2 + side3;
    }

    @Override
    double perimeter(){
        return 0.5 *(side1 * height);
    }
}

class AbstractClass1 {
    public static void main(String[] args){

        System.out.println("Name : Gokul Sarkar \nRoll No : 46");
        Rectangle rectangle = new Rectangle(15,20);
        Triangle triangle = new Triangle(5, 6, 4, 8);

        System.out.println("Area of Rectangle : "+rectangle.area());
        System.out.println("Perimeter of Rectangle : "+rectangle.perimeter());

        System.out.println("Area of Triangle : "+triangle.area());
        System.out.println("Perimeter of Triangle : "+triangle.perimeter());

    }
}
```

Output:

```
PS C:\Users\GOKUL SARKAR\Desktop\Java> javac AbstractClass1.java
PS C:\Users\GOKUL SARKAR\Desktop\Java> java AbstractClass1
Name : Gokul Sarkar
Roll No : 46
Area of Rectangle : 300.0
Perimeter of Rectangle : 70.0
Area of Triangle : 15.0
Perimeter of Triangle : 20.0
PS C:\Users\GOKUL SARKAR\Desktop\Java>
```

Name : Gokul Sarkar
Roll No : 46

2. Write a program to create a class named Vehicle having protected instance variables regnNumber, speed, colour, ownerName and a method showData ( ) to show “This is a vehicle class”. Inherit the Vehicle class into subclasses named Bus and Car having individual private instance variables routeNumber in Bus and manufacturerName in Car and both of them having showData ( ) method showing all details of Bus and Car respectively with the content of the super class’s showData ( ) method.

```
class Vehicle{
   protected int regnNumber, speed;
   protected String color, wonerName;

   void showData(){
      System.out.println("This is a Vehicle class....");
   }
}
class Bus extends Vehicle{
   private int routeNumber;
   Bus(int regnNumber, int speed, String color, String wonerName, int routeNumber){
      this.regnNumber = regnNumber;
      this.speed = speed;
      this.color = color;
      this.wonerName = wonerName;
      this.routeNumber = routeNumber;

   }

   void showData(){
      super.showData();
      System.out.println("Registration Numner : "+ regnNumber);
      System.out.println("Speed : "+speed);
      System.out.println("Color : "+ color);
      System.out.println("Woner name : "+ wonerName);
      System.out.println("routeNumber : "+ routeNumber);
   }
```

Name : Gokul Sarkar
Roll No : 46

```
}
class Car extends Vehicle{
   private String manufacturerName;
   Car(int regnNumber, int speed, String color, String wonerName,String
manufacturerName){
      this.regnNumber = regnNumber;
      this.speed = speed;
      this.color = color;
      this.wonerName = wonerName;
      this.manufacturerName = manufacturerName;
   }

   @Override
   void showData(){
      super.showData();
      System.out.println("Registration Numner : "+ regnNumber);
      System.out.println("Speed : "+speed);
      System.out.println("Color : "+ color);
      System.out.println("Woner name : "+ wonerName);
      System.out.println("manufacturerName : "+ manufacturerName);
   }

}

public class AbstractClass2 {
   public static void main(String[] args){
      System.out.println("Name : Gokul Sarkar \nRoll No : 46");
      Car car = new Car(120, 70, "Blue", "Gokul", "ABC");
      car.showData();

      System.out.println();

      Bus bus = new Bus(240, 60, "Yellow", "XYZ", 1944);
      bus.showData();

   }
}
```

Name : Gokul Sarkar
Roll No : 46

Output:

```
PS C:\Users\GOKUL SARKAR\Desktop\Java> javac AbstractClass2.java
PS C:\Users\GOKUL SARKAR\Desktop\Java> java AbstractClass2
Name : Gokul Sarkar
Roll No : 46
This is a Vehicle class....
Registration Numner : 120
Speed : 70
Color : Blue
Woner name : Gokul
manufacturerName : ABC

This is a Vehicle class....
Registration Numner : 240
Speed : 60
Color : Yellow
Woner name : XYZ
routeNumber : 1944
PS C:\Users\GOKUL SARKAR\Desktop\Java>
```

3. Create an interface Department containing attributes deptName and deptHead. It also has abstract methods for printing the attributes. Create a class hostel containing hostelName, hostelLocation and numberofRooms. The class contains methods for getting and printing the attributes. Then write a Student class extending the Hostel class and implementing the Department interface. This class contains attributes studentName, regdNo, electiveSubject and avgMarks. Write suitable getData and printData methods for this class. Also, implement the abstract methods of the Department interface. Write a driver class to test the Student class. The program should be menu driven containing the options:
i) Admit new student
ii) Migrate a student
iii) Display details of a student
For the third option, a search is to be made on the basis of the entered registration number.

Name : Gokul Sarkar
Roll No : 46

```
import java.util.Scanner;

interface Department{
   String deptName = "MCA";
   String deptHead = "Kaustuv Bhattacharjee";
   void displayDept();
}

class Hostel{
   Scanner scan = new Scanner(System.in);
   String hostelName;
   String hostelLocation;
   int numberOfRooms;

   void Attributes(){
      System.out.print("Hostel Name: ");
      this.hostelName = scan.nextLine();
      scan.nextLine();
      System.out.print("Hostel Location: ");
      this.hostelLocation = scan.nextLine();
      System.out.print("Number of Rooms: ");
      this.numberOfRooms = scan.nextInt();
   }

   void display(){
      System.out.println("Hostel Name is: " + this.hostelName);
      System.out.println("Hostel Location is at: " + this.hostelLocation);
      System.out.println("There are total : " + this.numberOfRooms + "
Rooms.");
   }
}

class Student extends Hostel implements Department{
   String studentName;
   long regNo;
   String electiveSubject;
   float avgMarks;
```

Name : Gokul Sarkar
Roll No : 46

```
    void values(){
        System.out.print("Student Name: ");
        studentName = scan.nextLine();
        System.out.print("Elective Subject is: ");
        electiveSubject = scan.nextLine();
        System.out.print("Registration Number: ");
        regNo = scan.nextLong();
        System.out.print("Average Marks is: ");
        avgMarks = scan.nextFloat();
    }

    void displayValues(){
        System.out.println("Student Name: " + studentName);
        System.out.println("Elective Subject: " + electiveSubject);
        System.out.println("Registration Number: " + regNo);
        System.out.println("Average Marks: " + avgMarks);
        System.out.println("Hostel Name: " + hostelName);
        System.out.println("Hostel Location: " + hostelLocation);
        System.out.println("Total Room Number: " + numberOfRooms);
    }

    @Override
    public void displayDept() {
        System.out.println("Department Name: " + deptName);
        System.out.println("Department Head: " + deptHead);
    }
}

public class AbstractClass3 {
    public static void main(String[] args) {
        System.out.println("Name : Gokul Sarkar \nRoll No : 46");
        Student s = new Student();
        s.values();
        s.Attributes();
        s.displayValues();
        s.displayDept();
    }
}
```

Name : Gokul Sarkar
Roll No : 46

Output:

```
PS C:\Users\GOKUL SARKAR\Desktop\Java> javac AbstractClass3.java
PS C:\Users\GOKUL SARKAR\Desktop\Java> java AbstractClass3
Name : Gokul Sarkar
Roll No : 46
Student Name: Gokul Sarkar
Elective Subject is: ABCD
Registration Number: 2580
Average Marks is: 75
Hostel Name: XYZ
Hostel Location: Newtown
Number of Rooms: 15
Student Name: Gokul Sarkar
Elective Subject: ABCD
Registration Number: 2580
Average Marks: 75.0
Hostel Name:
Hostel Location: Newtown
Total Room Number: 15
Department Name: MCA
Department Head: Kaustuv Bhattacharjee
PS C:\Users\GOKUL SARKAR\Desktop\Java>
```

4. Create an abstract class Accounts with the following details:
Data Members:
- **(a) Balance**
- **(b) accountNumber**
- **(c) accountHoldersName**
- **(d) address**

Methods:
- **(a) withdrawl()- abstract**
- **(b) deposit()- abstract**
- **(c) display() to show the balance of the account number**

Create a subclass of this class SavingsAccount and add the following details:
Data Members:
- **(a) rateOfInterest**

Methods:
- **(a) calculateAount()**

Name : Gokul Sarkar
Roll No : 46

```
abstract class Accounts{
   int balance =0;
   int accountNumber;
   String accHolderName;
   String address;

   abstract void withdrawl(int amount);
   abstract void deposit(int amount);

   void display(){
      System.out.println("Account number : "+ accountNumber);
      System.out.println("Balance : "+ balance);
   }

}

class SavingAccount extends Accounts{
   float rateOfInterest;
   public SavingAccount(int accountNumber,String accHolderName, String
address,float rateOfInterest){
      this.accHolderName = accHolderName;
      this.accountNumber = accountNumber;
      this.address = address;
      this.rateOfInterest = rateOfInterest;
   }

   @Override
   public void withdrawl(int amount){
      if(balance>+amount){
         balance -= amount;
         System.out.println(amount+" withdrawl sucessfully...");
         System.out.println("balance : "+ balance);
      }
      else{
         System.out.println("Insufficient balance....");
      }
```

Name : Gokul Sarkar
Roll No : 46

```
    }
    public void deposit(int amount){
        balance += amount;
        System.out.println("total amount : "+balance);
    }
    void calculateAmount(int year){
        float interest = balance * rateOfInterest * year / 100;
        balance += interest;
        System.out.println("Total amount after interest : "+ balance);
    }
}

public class AbstractClass4 {
    public static void main(String[] args) {
        System.out.println("Name : Gokul Sarkar \nRoll No : 46");
        SavingAccount sa = new SavingAccount(145, "Gokul", "Balurghat", 8);
        sa.deposit(2000);
        sa.display();
        sa.withdrawl(100);
        sa.display();
        sa.calculateAmount(5);
    }
}
```

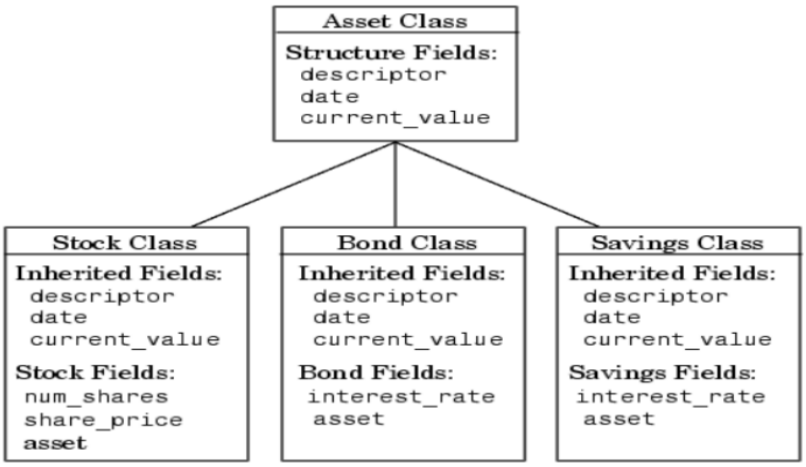
Output:

```
PS C:\Users\GOKUL SARKAR\Desktop\Java> javac AbstractClass4.java
PS C:\Users\GOKUL SARKAR\Desktop\Java> java AbstractClass4
Name : Gokul Sarkar
Roll No : 46
total amount : 2000
Account number : 145
Balance : 2000
100 withdrawl sucessfully...
balance : 1900
Account number : 145
Balance : 1900
Total amount after interest : 2660
PS C:\Users\GOKUL SARKAR\Desktop\Java>
```

Name : Gokul Sarkar
Roll No : 46

5. Implement the below Diagram.

Here, Asset class is an abstract class containing an abstract method displayDetails() method. Stock, bond and Savings class inherit the Asset class and displayDetails() method is defined in every class.




```
abstract class Asset{
   String descriptor;
   String date;
   int current_value;
   Asset(String a, String b, int c){
      descriptor =a;
      date= b;
      current_value= c;
   }
   abstract void displayDetails();
}
class Stock extends  Asset{
   int num_share;
   int share_price;
   String asset;
   Stock(String a, String b, int c, int d, int e, String f){
      super(a,b,c);
      num_share=d;
      share_price=e;
      asset=f;
   }
   void displayDetails(){
```

Name : Gokul Sarkar
Roll No : 46

```
      System.out.println("Details of Stocks is:"+"\n"+"Name: "+descriptor+"\n"+"Date of Purchase:"+date+"\n"+"Value of each: "+current_value+"\n"+"No. of Shares: "+num_share+"\n"+"Initial Price: "+share_price+"\n"+"Type: "+asset);
   }
}
class Bond extends  Asset{
   double interestrate;
   String asset;
   Bond(String a, String b, int c, double g, String h){
      super(a,b,c);
      interestrate=g;
      asset=h;
   }
   void displayDetails(){
      System.out.println("Details of Bonds is:"+"\n"+"Name: "+descriptor+"\n"+"Date of Purchase: "+date+"\n"+"Value: "+current_value+"\n"+"INTEREST RATE: "+interestrate+"\n"+"Type: "+asset);
   }
}
class Savings extends  Asset{
   double interestrate;
   String asset;
   Savings(String a, String b, int c, double x, String y){
      super(a,b,c);
      interestrate=x;
      asset=y;
   }
   void displayDetails(){
      System.out.println("Details of Savings is:"+"\n"+"Name: "+descriptor+"\n"+"Date of Opening: "+date+"\n"+"Current Balance: "+current_value+"\n"+"INTEREST RATE: "+interestrate+"\n"+"Type: "+asset);
   }
}
class AbstractClass5 {
   public static void main(String[] args){
      System.out.println("Name : Gokul Sarkar \nRoll No : 46");
```

Name : Gokul Sarkar
Roll No : 46

```
    Bond obj1=new Bond("Private Bond","19/06/1998",200000,8.0,"Low Risk High Yield");
    Stock obj2=new Stock("Tiffany & co.","20/02/2005",1002,10,90,"High Risk High Yield");
    Savings obj3= new Savings("Single Account", "10/02/1996",1000000,5.5,"Low Risk Low Yield");
    obj2.displayDetails();
    System.out.println("********");
    obj1.displayDetails();
    System.out.println("********");
    obj3.displayDetails();
    System.out.println("********");
  }
}
```

Output:

```
PS C:\Users\GOKUL SARKAR\Desktop\Java> javac AbstractClass5.java
PS C:\Users\GOKUL SARKAR\Desktop\Java> java AbstractClass5
Name : Gokul Sarkar
Roll No : 46
Details of Stocks is:
Name: Tiffany & co.
Date of Purchase:20/02/2005
Value of each: 1002
No. of Shares: 10
Initial Price: 90
Type: High Risk High Yield
********
Details of Bonds is:
Name: Private Bond
Date of Purchase: 19/06/1998
Value: 200000
INTEREST RATE: 8.0
Type: Low Risk High Yield
********
Details of Savings is:
Name: Single Account
Date of Opening: 10/02/1996
Current Balance: 1000000
INTEREST RATE: 5.5
Type: Low Risk Low Yield
********
PS C:\Users\GOKUL SARKAR\Desktop\Java>
```