ABSTRACT CLASS

Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.

abstract class Parent{

    abstract void message();

}

class Subclass1 extends Parent{

    public void message(){

        System.out.println("This is first subclass");

    }

}

class Subclass2 extends Parent{

    public void message(){

        System.out.println("This is Second subclass");

    }

}

class Demo1{

    public static void main(String[] args){

        Subclass1 s1 = new Subclass1();

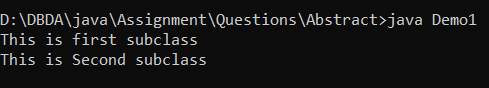
        s1.message();

        Subclass2 s2 = new Subclass2();

        s2.message();

    }

}



2.

Create an abstract class 'Bank' with an abstract method 'getBalance'. $100, $150 and $200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.

abstract class Bank{

    abstract void getBalance();

}

class BankA extends Bank{

    public void getBalance(){

        System.out.println("Balence in a bank A is: $100 ");

    }

}

class BankB extends Bank{

    public void getBalance(){

        System.out.println("Balence in a bank B is: $150 ");

    }

}

class BankC extends Bank{

    public void getBalance(){

        System.out.println("Balence in a bank C is: $200");

    }

}

class Demo2{

    public static void main(String[] args){

        BankA a = new BankA();

        BankB b = new BankB();

        BankC c = new BankC();

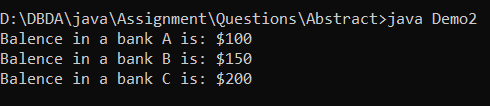
        a.getBalance();

        b.getBalance();

        c.getBalance();

    }

}



3.

We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for eac of the two classes and print the percentage of marks for both the students.

abstract class Marks{

    abstract void getPercentage();

}

class A extends Marks{

    int m1;

    int m2;

    int m3;

    A(int m1, int m2, int m3){

        this.m1 = m1;

        this.m2 = m2;

        this.m3 = m3;

    }

    void getPercentage(){

        float avg;

        avg = (m1+m2+m3)/3;

        System.out.println("Avg marks of student A is: "+avg);

    }

}

class B extends Marks{

    int m1;

    int m2;

    int m3;

    int m4;

    B(int m1, int m2, int m3, int m4){

        this.m1 = m1;

        this.m2 = m2;

        this.m3 = m3;

        this.m4 = m4;

    }

    void getPercentage(){

        float avg;

        avg = (m1+m2+m3+m4)/4;

        System.out.println("Avg marks of student B is: "+avg);

    }

}

class Demo3{

    public static void main(String[] args){

        A a = new A(75,96,64);

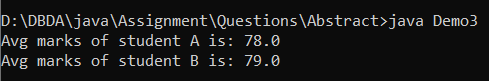
        B b = new B(85,63,78,92);

        a.getPercentage();

        b.getPercentage();

    }

}



4.

An abstract class has a construtor which prints "This is constructor of abstract class", an abstract method named 'a\_method' and a non-abstract method which prints "This is a normal method of abstract class". A class 'SubClass' inherits the abstract class and has a method named 'a\_method' which prints "This is abstract method". Now create an object of 'SubClass' and call the abstract method and the non-abstract method. (Analyse the result)

//abstract class

abstract class Cls{

    //abstract method

    abstract void a\_method();

    //non-abstract method

    void n\_method(){

        System.out.println("This is a normal method of abstract class");

    }

    Cls(){

        System.out.println("This is constructor of abstract class");

    }

}

class SubClass extends Cls{

    public void a\_method(){

        System.out.println("This is abstract method");

    }

}

//main class

class Demo4{

    public static void main(String[] args){

        //create an object

        SubClass s = new SubClass();

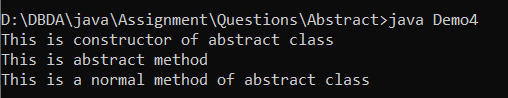
        //call a method

        s.a\_method();

        s.n\_method();

    }

}



5.

Create an abstract class 'Animals' with two abstract methods 'cats' and 'dogs'. Now create a class 'Cats' with a method 'cats' which prints "Cats meow" and a class 'Dogs' with a method 'dogs' which prints "Dogs bark", both inheriting the class 'Animals'. Now create an object for each of the subclasses and call their respective methods.

abstract class Animals{

    abstract void cats();

    abstract void dogs();

}

class Cats extends Animals{

    public void cats(){

        System.out.println("Cats meow");

    }

    void dogs(){

    }

}

class Dogs extends Animals{

    public void dogs(){

        System.out.println("Dogs bark");

    }

    void cats(){

    }

}

class Demo5{

    public static void main(String[] args){

        Cats c = new Cats();

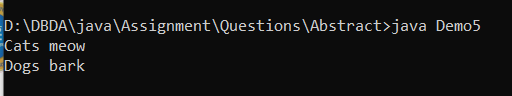
        Dogs d = new Dogs();

        c.cats();

        d.dogs();

    }

}



6.

We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.

abstract class Shape{

    abstract void rectangleArea(int length, int breadth);

    abstract void squareArea(int side);

    abstract void circleArea(double radius);

}

class Area{

    void rectangleArea(int length, int breadth){

        int rect = length \* breadth;

        System.out.println("Are of Rectangle is: " +rect);

    }

    void squareArea(int side){

        int sqr = side \* side;

        System.out.println("Are of Square is: " +sqr);

    }

    void circleArea(double radius){

        double cir = 3.14 \* radius \* radius;

        System.out.println("Area of Square is: " +cir);

    }

    }

class Demo6{

    public static void main(String[] args){

        Area a = new Area();

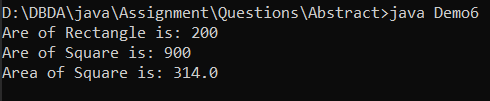
        a.rectangleArea(10,20);

        a.squareArea(30);

        a.circleArea(10);

    }

}



7.

Repeat the above question for 4 rectangles, 4 squares and 5 circles.

Hint- Use array of objects.