Q1. Write java program to demonstrate implementation of abstract class.

**Code:**

abstract class Clit {

public abstract int sqrt(int n1);

public abstract int cube(int n1);

public void show() {

System.out.println("Im hareesh");

}

}

public class Pract4c extends Clit {

@Override

public int sqrt(int n1){

return(n1\*n1);

}

@Override

public int cube(int n1){

return(n1\*n1\*n1);

}

public static void main(String[] args) {

Pract4c obj = new Pract4c();

int sq = obj.sqrt(28);

System.out.println("28 Square = " + sq);

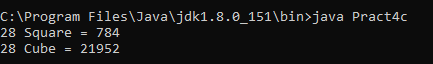
int cube = obj.cube(28);

System.out.println("28 Cube = " + cube);

}

}

**Output:**



Q2. Write java program to implement single level inheritance.

**Code:**

class demo {

float pi = 3.14f;

void show() {

System.out.println("Area of circle: ");

}

}

public class Pract5a extends demo {

float r = 2.0f;

void area() {

System.out.println(pi\*r\*r);

}

public static void main(String[] args) {

Pract5a obj = new Pract5a();

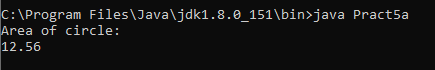
obj.show();

obj.area();

}

}

**Output:**



Q3. How interface are implemented with Annotations.

**Code:**

@FunctionalInterface

interface Drawable {

public void age();

}

public class Hareesh{

public static void main(String[] args) {

int age = 19;

Drawable ddd = () -> {

System.out.println("Hareesh age: " + age);

};

ddd.age();

}

}

**Output:**

****

Q4. Write java program to implement method overriding.

**Code:**

class A {

void show(){

System.out.println("Im hareesh");

}

}

class B extends A {

void show(){

System.out.println("Im also hareesh");

}

}

public class Pract5b{

public static void main(String[] args) {

B ob = new B();

ob.show();

}

}

**Output:**

