FuLLSTACK ASSIGNMENT 3

**Question no**.**1: What is Inheritance?**

**Answer** **:** **Inheritance** is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class.

**Question no.2: What is Multiple Inheritance?**

**Answer** **:** When one class extends more than one classes then this is called **multiple inheritance**. For example: Class C extends class A and B then this type of inheritance is known as multiple inheritance.

**Question no.3: What is the use of super keyword?**

**Answer** **:** The super keyword refers to superclass (parent) objects. It is used to call superclass methods, and to access the superclass constructor. The most common use of the super keyword is to eliminate the confusion between superclasses and subclasses that have methods with the same name.

**Question no.4: What is abstract method?**

**Answer** **:** A method which is declared as abstract and does not have implementation is known as an abstract method**. Example:** **abstract** **void** printStatus();(here it does not have any method body and abstract).

**Question no.5: What is Abstract class?**

**Answer** **:** A class which is declared with the abstract keyword is known as an abstract class in Java. It can have abstract and non-abstract methods (method with the body).

**Question no.6: What is use of final modifier?**

**Answer** **:** The final is a modifier in Java, which can be applied to a variable, a method or a class. When a final modifier is used with a class then the class cannot be extended further. This is one way to protect your class from being subclassed and often sensitive classes are made final due to security reasons.

**Question no**.**7: What is Interface? Write the syntax of interface.**

**Answer** **:** An Interface in Java programming is defined as an abstract type used to specify the behavior of a class. A Java interface contains static constants and abstract methods. A class can implement multiple interfaces. In Java, interfaces are declared using the interface keyword.

General syntax:

interface {

methods

}

**Question no.8: What is package?**

**Answer** **:** A **java package** is a group of similar types of classes, interfaces and sub-packages. Package in java can be categorized in two form, built-in package and user-defined package. There are many built-in packages such as java, lang, javax, io, util etc.

**Question no.9: What is Exception?**

**Answer** **:** An exception (or exceptional event) is a problem that arises during the execution of a program. When an **Exception** occurs the normal flow of theprogram is disrupted and the program/Application terminates abnormally, which is not recommended, therefore, these exceptions are to be handled. An exception can occur for many different reasons. Example: A user has entered an invalid data.

**Question no.10: What is the use of finally block?**

**Answer** **:** Java finally block is a block that is used to execute important code such as closing connection, stream etc.

Java finally block is always executed whether exception is handled or not. Java finally block follows try or catch block**.**

Programs

1. **Create a class Publication with data members title(String) and price(int). From this class derive two classes Book and CD. Class Book adds pages(int) and CD adds Size(int). Each of these classes should have constructors and display(). Write a java program to implement this using super, this and method overriding concepts.**

**Sol:**

class Publications{

String title="Harry Potter";

int price=799;

int pages;

int size;

}

class Book extends Publications{

public Book(int pages)

{

this.pages=pages;

}

public void display(){

System.out.println("The price of the Book "+super.title+" is "+super.price+" with "+pages+" pages");

}

}

class CD extends Publications{

public CD(int size)

{

this.size=size;

}

public void display()

{

System.out.println("The price of the CD "+super.title+" is "+super.price+" with size "+size);

}

}

class bookcd

{

public static void main(String args[])

{

Book x=new Book(444);

CD y=new CD(27);

x.display();

y.display();

}

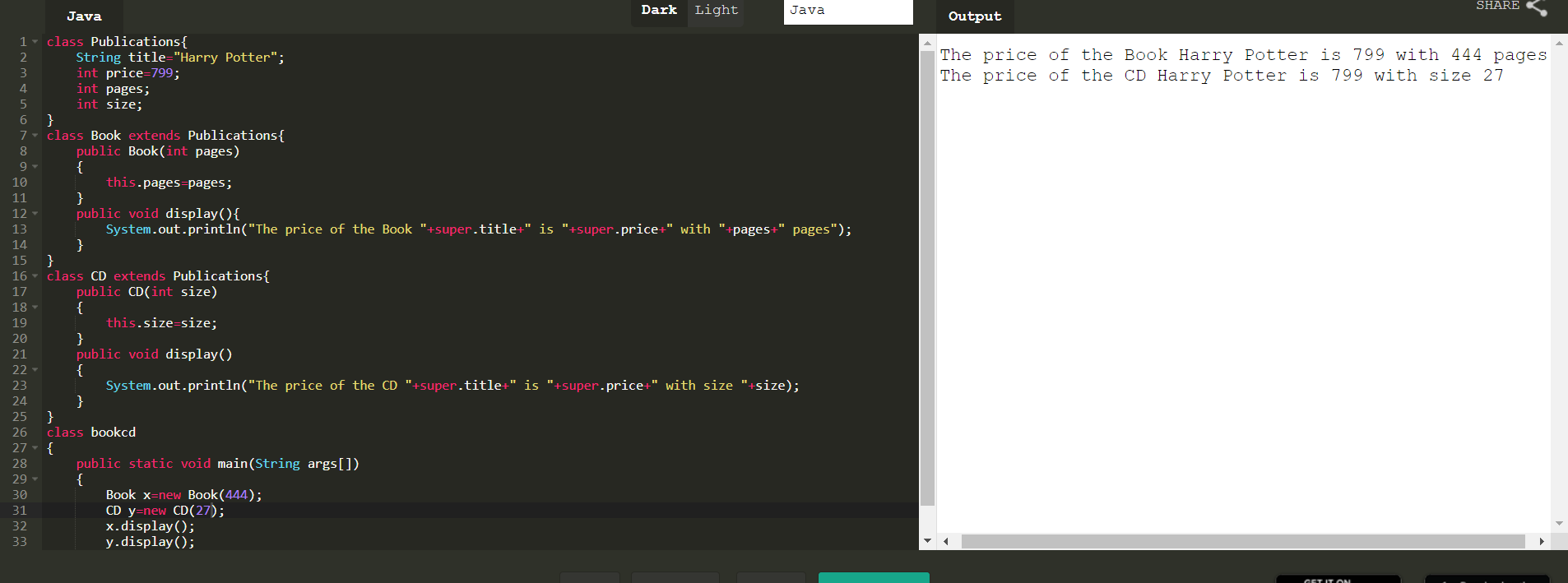
}

**Expected Output:**

The price of the Book Harry Potter is 799 with 444 pages

The price of the CD Harry Potter is 799 with size 27

**Output**



1. **Write a simple java program to demonstrate method overriding.**

**Sol:**

class A

{

void show()

{

System.out.println("This is class A-Overridden Method");

}

}

class B extends A

{

void showB()

{

System.out.println("This is class B-Overiding Method");

}

}

class ABC

{

public static void main(String []args)

{

B y=new B();

y.show();

y.showB();

}

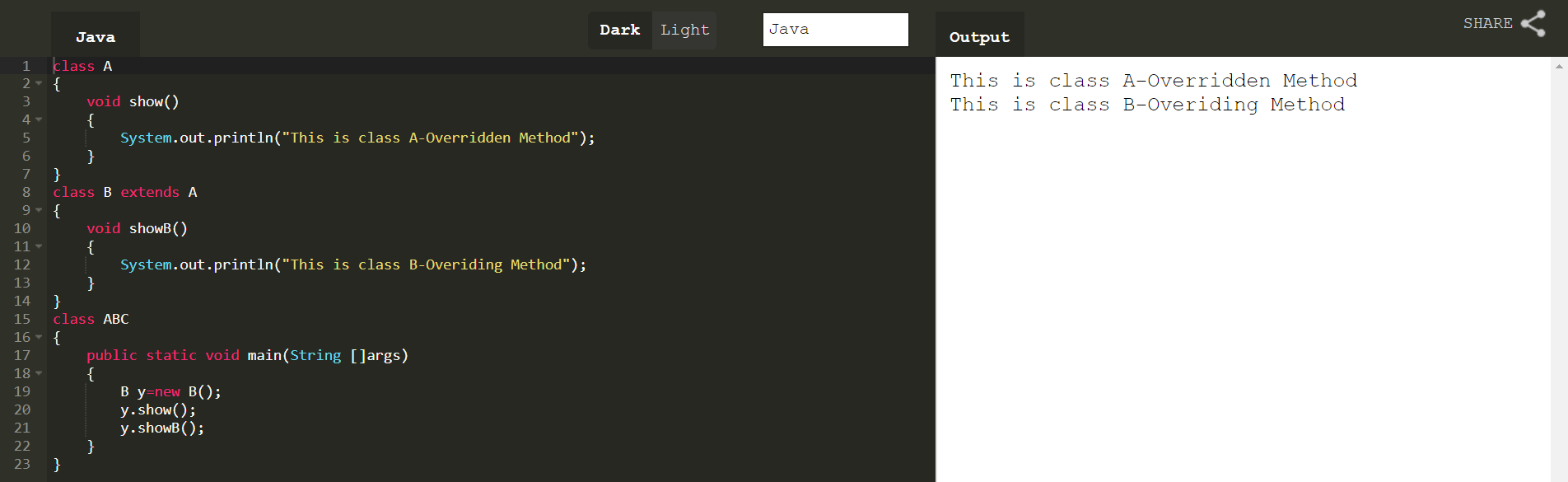
}

**Expected Output:**

This is class A-Overridden Method

This is class B-Overiding Method

**Output**



1. **Write a java program to create an interface called Shape with CalculateArea(). Create three classes namely Square,Circle,Triangle which implements Shape.**

**Sol:**

interface Shape

{

void CalculateArea();

}

class Circle implements Shape

{

int r = 5;

double pi = 3.14, ar ;

public void CalculateArea()

{

ar = pi \* r \* r;

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

}

}

class Square implements Shape

{

int s=4;

double ar;

public void CalculateArea()

{

ar = s\*s;

System.out.println("Area of square:"+ar);

}

}

class Triangle implements Shape

{

int b=10,h=5;

double ar;

public void CalculateArea()

{

ar = 0.5\*b\*h;

System.out.println("Area of triangle:"+ar);

}

}

public class findingarea

{

public static void main(String[] args)

{

Square x = new Square();

Circle y = new Circle();

Triangle z= new Triangle();

x.CalculateArea();

y.CalculateArea();

z.CalculateArea();

}

}

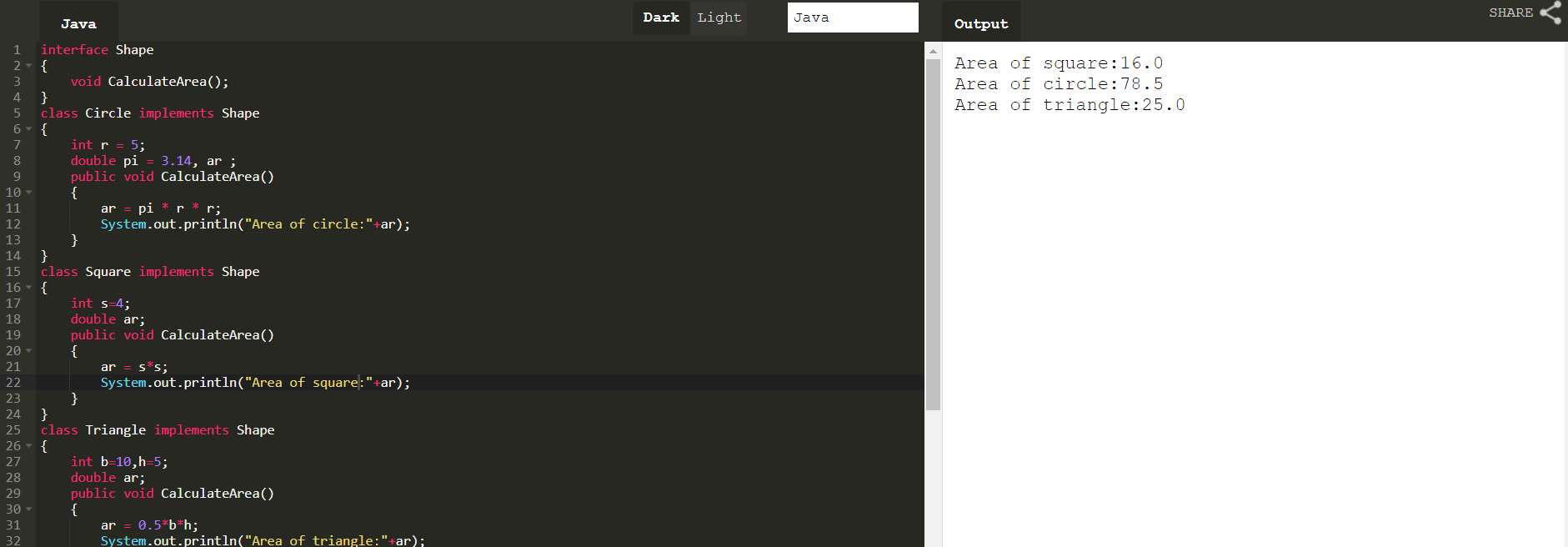
**Expected Output:**

Area of square:16.0

Area of circle:78.5

Area of triangle:25.0

**Output**



**4.Create two packages p1 and p2. The package p1 contains class A which contains one display(). Create class B in package p2. The main method of class B invoke A’s display(). Write a java program to do this.**

**Sol:**

package p1;

public class A

{

public void display()

{ System.out.println("I'm Bharathi");

}

}

package p2;

import p1.\*;

class B

{ public static void main(String args[])

{

A x = new A();

x.display();

}

}

**Expected Output:**

I’m Bharathi

**5.Write a java program to count numbers, characters in the command line arguments using Exception handling mechanism.**

**Sol:**

public class Commandline

{

public static void main(String[] args)

{

String s=args[0];

try{

int d=0,c=0;

for(int i=0;i<s.length();i++)

{

if(Character.isLetter(s.charAt(i)))

c++;

else

d++;

}

System.out.println("Number of Characters: "+c);

System.out.println("Number of Digits: "+d);

}

catch(Exception e)

{

System.out.println(e);

}

}

}

**Expected Output:**

Number of Characters: 0

Number of Digits: 1

**Output**

