Assignment-3

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.

Program:class Publication{

String title;

int price;

Publication(String title,int price)

{

this.title=title;

this.price=price;

}

public void display()

{

System.out.println("title : "+title+" price : "+price);

}

}

class Book extends Publication

{

int pages;

Book(int pages)

{

super("maths",500);

this.pages=pages;

}

public void display()

{

super.display();

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

}

}

class CD extends Publication

{

int size;

CD(int size)

{

super("science",600);

this.size=size;

}

public void display()

{

super.display();

System.out.println("size : "+size);

}

public static void main(String[] args) {

CD cd=new CD(10);

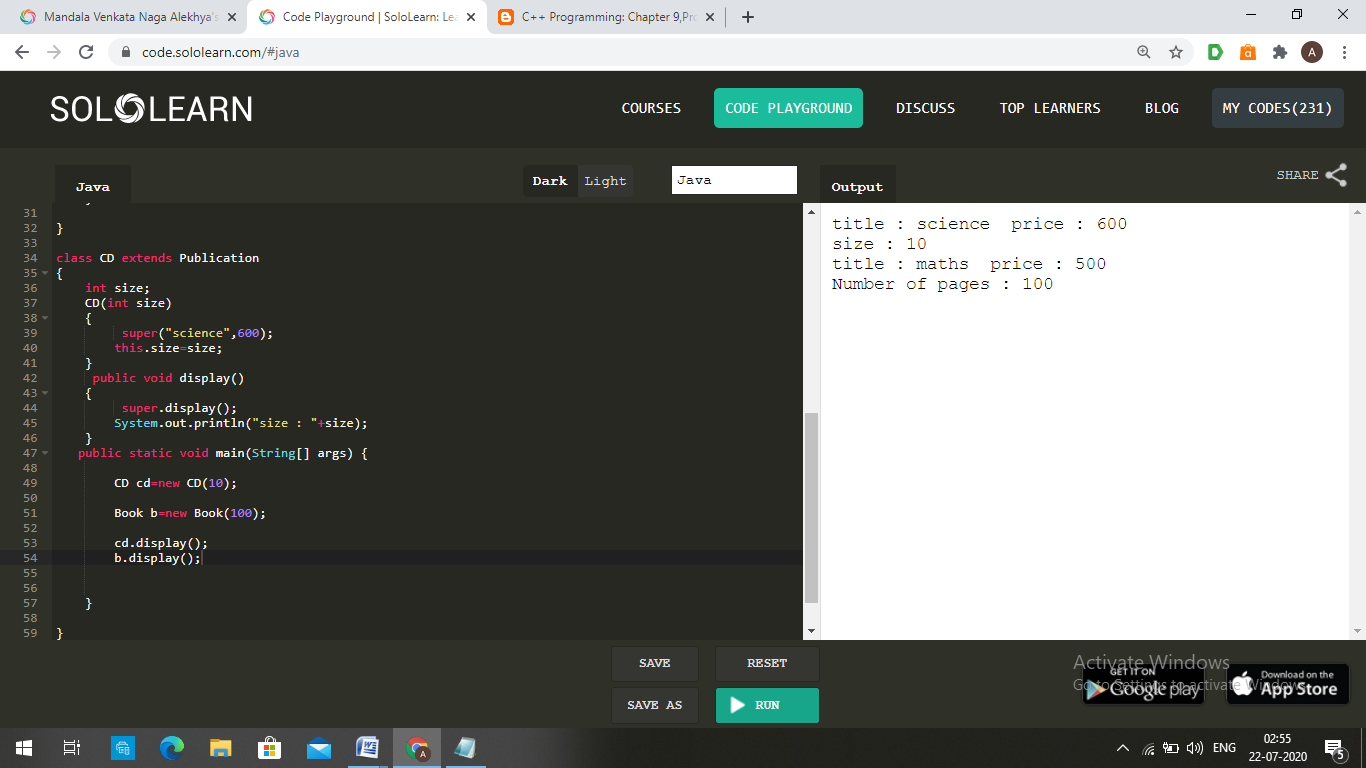
Book b=new Book(100);

cd.display();

b.display();

}}

Output:



2. Write a simple java program to demonstrate method overriding.

Program:class Human{

public void eat()

{

System.out.println("Human is eating");

}

}

class Boy extends Human{

public void eat(){

System.out.println("Boy is eating");

}

public static void main( String args[]) {

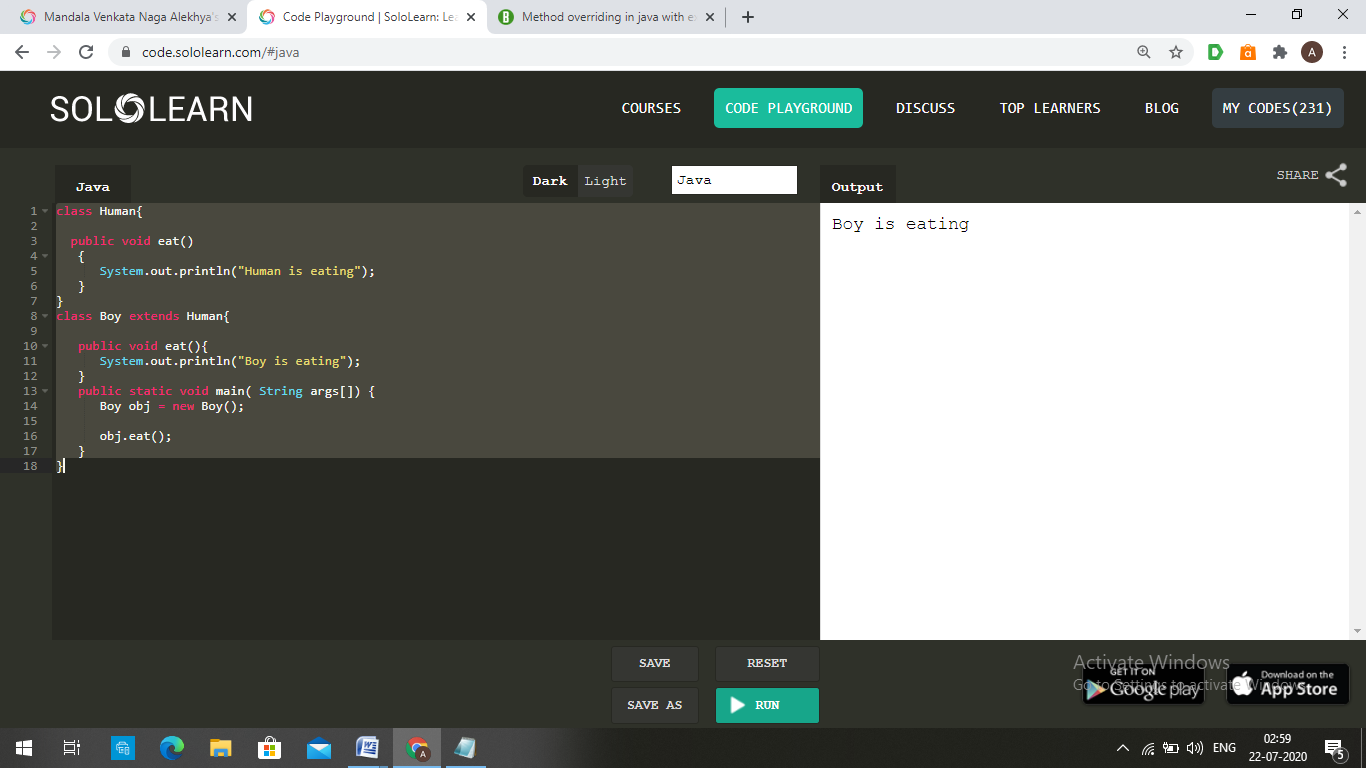
Boy obj = new Boy();

obj.eat();

}

}

Output:



3. Write a java program to create an interface called Shape with CalculateArea(). Create three

classes namely Square,Circle,Triangle which implements Shape.

Program: interface Shape

{

public void CalculateArea();

}

class Square implements Shape

{

int s;

Square(int s)

{

this.s=s;

}

public void CalculateArea()

{

System.out.println("Square area is "+s\*s);

}

}

class Circle implements Shape

{

int r;

Circle(int r)

{

this.r=r;

}

public void CalculateArea()

{

System.out.println("Circle area is "+(3.14\*r\*r));

}

}

class Triangle implements Shape

{

int b,h;

Triangle(int b,int h)

{

this.b=b;

this.h=h;

}

public void CalculateArea()

{

System.out.println("Triangle area is "+(0.5\*b\*h));

}

public static void main(String args[])

{

Square s=new Square(5);

Circle c=new Circle(4);

Triangle t=new Triangle(5,6);

s.CalculateArea();

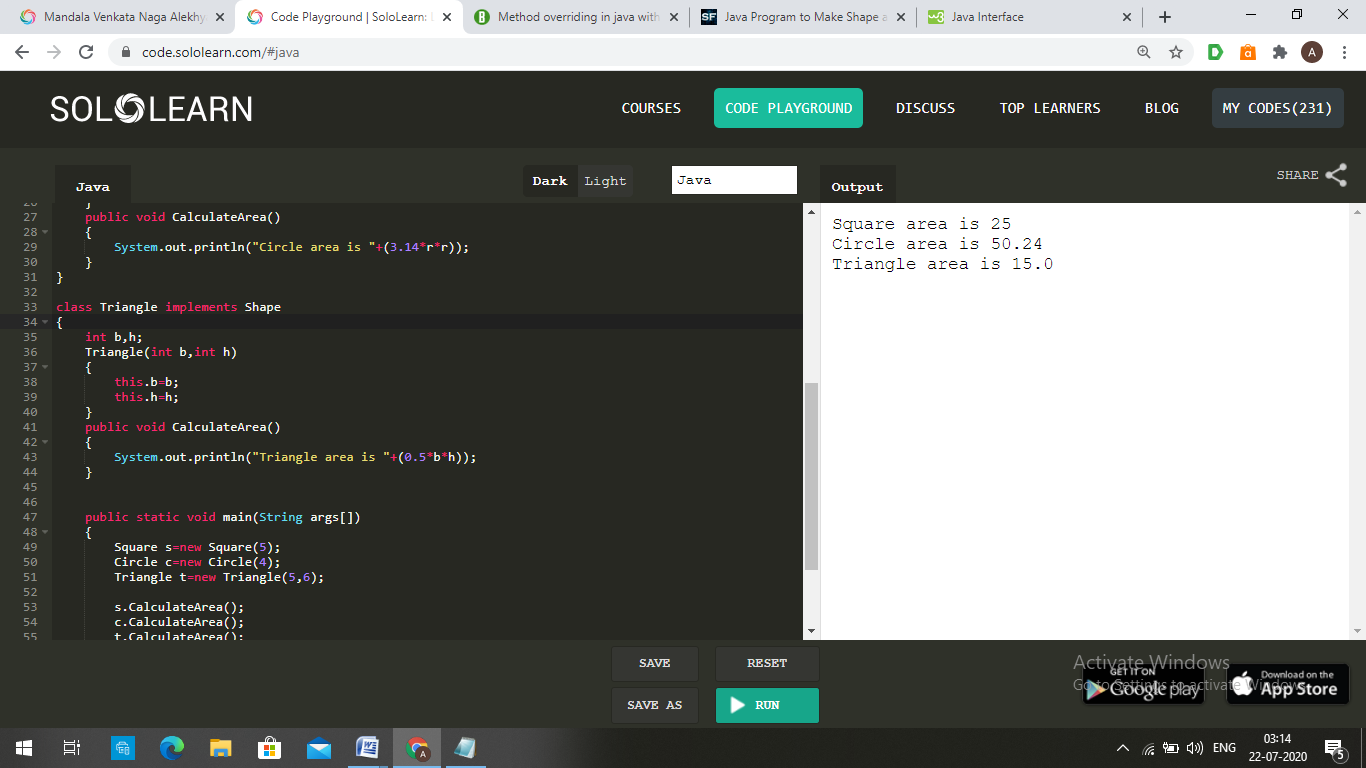
c.CalculateArea();

t.CalculateArea();

}

}

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.

Program: package p1;

public class A {

public void display()

{

System.*out*.println("Hello, I am from class A");

}

}

package p2;

import p1.A;

public class B {

public static void main(String args[])

{

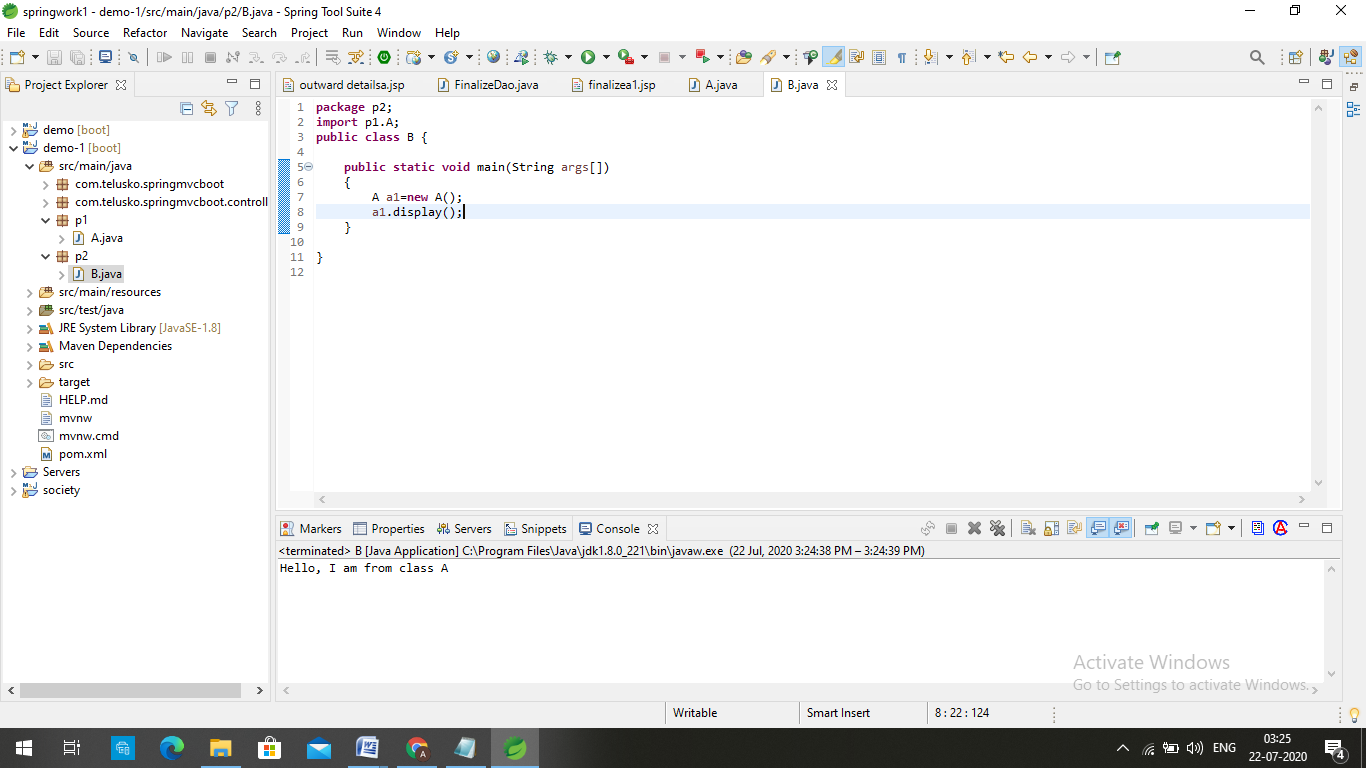
A a1=new A();

a1.display();

}

}

Output:



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

Exception handling mechanism.

Program: class Command

{

public static void main(String args[])

{

Try

{

String s1=args[0];

String s2=args[1];

int c1=0,c2=0;

for(int i=0;i<s1.length();i++)

{

if(Character.isDigit(s1.charAt(i))) c1+=1;

else if(Character.isLetter(s1.charAt(i)))c2+=1;

}

System.out.println("number of numbers in string1 is : "+c1+" , number of characters in string1 is : "+c2);

c1=0;c2=0;

for(int i=0;i<s2.length();i++)

{

if(Character.isDigit(s2.charAt(i))) c1+=1;

else if(Character.isLetter(s2.charAt(i)))c2+=1;

}

System.out.println("number of numbers in string2 is : "+c1+" , number of characters in string2 is : "+c2);

}

catch(ArrayIndexOutOfBoundsException e)

{

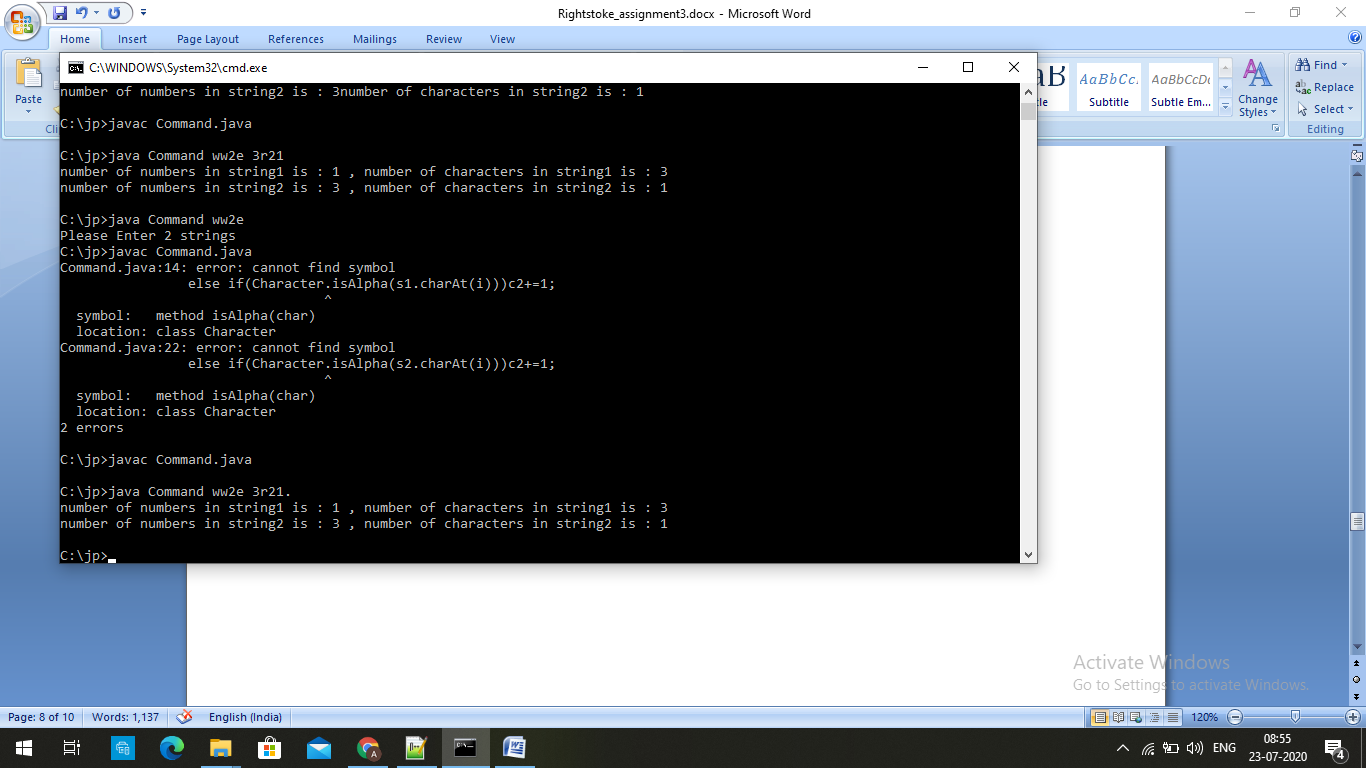
System.out.print("Please Enter 2 strings");

}

}

}

Output:



Theory

1. What is Inheritance?

Answer: Inheritance can be defined as the process where one class acquires the properties

(methods and fields) of another.

The class which inherits the properties of other is known as subclass (derived class,

child class) and the class whose properties are inherited is known as superclass (base

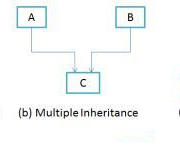
class, parent class).

2. What is Multiple Inheritance?

Answer: **“Multiple Inheritance”** refers to the concept of one class extending (Or inherits) more

than one base class.. The problem with “multiple inheritance” is that the derived class

will have to manage the dependency on two base classes.

[](https://beginnersbook.com/wp-content/uploads/2013/05/Multiple-Inheritance.png)

3. What is the use of Super keyword?

Answer: The super keyword in Java is a reference variable which is used to refer immediate

parent class object.

Whenever you create the instance of subclass, an instance of parent class is created

implicitly which is referred by super reference variable.

Usage of Java super Keyword

1. super can be used to refer immediate parent class instance variable.
2. super can be used to invoke immediate parent class method.
3. super() can be used to invoke immediate parent class constructor.

4. What is abstract method?

Answer: A method without body (no implementation) is known as abstract method. A method

must always be declared in an abstract class, or in other words you can say that if a class

has an abstract method, it should be declared abstract as well.

5. What is abstract class?

Answer: A class that is declared using “abstract” keyword is known as abstract class. It can

have abstract methods(methods without body) as well as concrete methods (regular

methods with body).

An abstract class can not be instantiated, which means you are not allowed to create an

object of it.

6. What is the 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 reason.

* When the final keyword is used with a method that it cannot be overridden in

Java, which means you cannot override the logic of the method in the subclass.

* When the final keyword is used with a variable then its value cannot be changed

once assigned.

7. What is interface? Write the syntax interface.

Answer: An **interface in Java** is a blueprint of a class. It has static constants and abstract

methods.

The interface in Java is a mechanism to achieve [abstraction](https://www.javatpoint.com/abstract-class-in-java). There can be only abstract

methods in the Java interface, not method body. It is used to achieve abstraction and

multiple [inheritance in Java](https://www.javatpoint.com/inheritance-in-java).

Syntax:

interface <interface\_name>{

// declare constant fields

// declare methods that abstract

// by default.

}

8. What is package?

Answer: A package as the name suggests is a pack(group) of classes, interfaces and other

packages. In java we use packages to organize our classes and interfaces. We have

two **types of packages in Java:** built-in packages and the packages we can create (also

known as user defined package).

9. What is exception?

Answer: Exception is an error event that can happen during the execution of a program and

disrupts it’s normal flow. Exception can arise from different kind of situations such as

wrong data entered by user, hardware failure, network connection failure etc.

Whenever any error occurs while executing a java statement, an exception object is

created and then [JRE](https://www.journaldev.com/546/difference-jdk-vs-jre-vs-jvm) tries to find exception handler to handle the exception.

10. What is the use of finally block?

Answer: The finally block will execute when the try/catch block leaves the execution, no matter

what condition cause it. **It always executes whether the try block terminates normally or**

**terminates due to an exception.** The main purpose of finally block is to release the

system resources. The finally block follows try/catch block.