**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.
2. Write a simple java program to demonstrate method overriding.
3. Write a java program to create an interface called Shape with CalculateArea(). Create three classes namely Square,Circle,Triangle which implements Shape.
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.
5. Write a java program to count numbers, characters in the command line arguments using Exception handling mechanism.

1. What is Inheritance?

2. What is Multiple Inheritance?

3. What is the use of Super keyword?

4. What is abstract method?

5. What is abstract class?

6. What is the use of final modifier?

7. What is interface? Write the syntax interface.

8. What is package?

9. What is exception?

10. What is the use of finally block?

ANSWERS

1)

import java.util.\*;

public class Publication{

int price;

String title;

public Publication(int price,String title){

this.price=price;

this.title=title;

}

}

public class book extends Publication

{

int pages;

public book(int price,String title,int pages){

super(price,title);

this.pages=pages;

}

void display(){

System.out.println("The price of the book is: "+price);

System.out.println("The title of the book is : "+title);

System.out.println("Total number of pages are: "+this.pages);

}

}

public class CD extends book

{

int size;

public CD(int price,String title,int pages,int size){

super(price,title,pages);

this.size=size;

}

void display(){

System.out.println("The price of the book is: "+price);

System.out.println("The title of the book is : "+title);

System.out.println("Total number of pages are: "+pages);

System.out.println("The total size of the book is :"+this.size+"sqcm2");

}

}

public class Program1

{

public static void main(String[] args) {

book p1=new book(200,"Advanced Java Handbook",450);

CD p2=new CD(200,"Advanced Java Handbook",450,6);

p1.display();

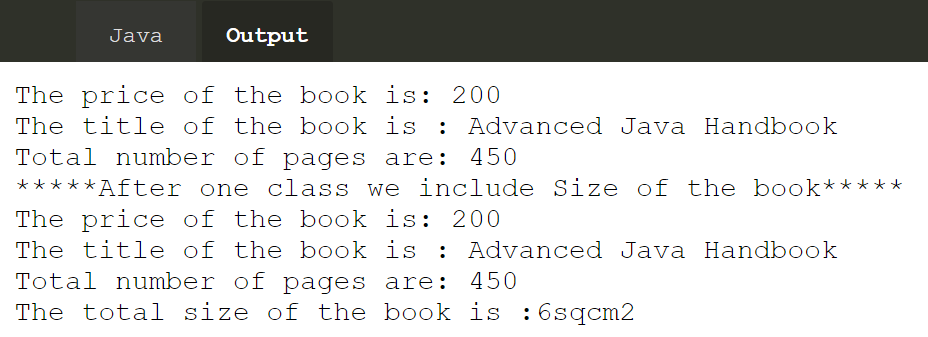
System.out.println("\*\*\*\*\*After one class we include Size of the book\*\*\*\*\*");

p2.display();

}

}

Output:



2)

class classRoom

{

void run(){

System.out.println("method Overriding to extend a class from parent class to child class");

System.out.println("This is online classroom");

}

}

class Program2 extends classRoom

{

public static void main(String[] args) {

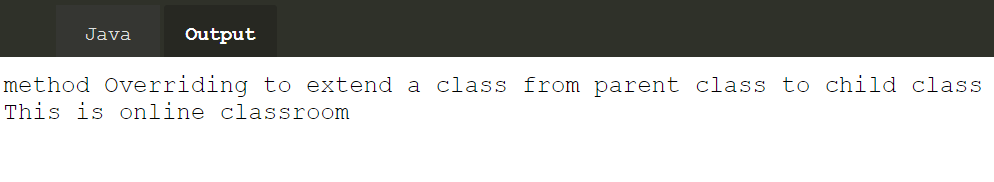
Program2 b1=new Program2();

b1.run();

}

}

Output:



3)

interface Drawable

{

void draw();

}

public class Circle implements Drawable

{

public void draw(){

System.out.println("Now Circle is creating");

}

}

public class Square implements Drawable

{

public void draw(){

System.out.println("Now Square is creating");

}

}

public class Triangle implements Drawable

{

public void draw(){

System.out.println("Now Triangle is creating");

}

}

public class Program3

{

public static void main(String[] args) {

Drawable c=new Circle();

c.draw();

Drawable s=new Square();

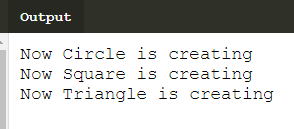
s.draw();

Drawable t=new Triangle();

t.draw();

}

}

Output:  


5)

import java.util.\*;

class Count

{

public static void main(String a[])

{

String s=a[0];

try

{

int digit=0;

int chr=0;

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

{

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

chr++;

else

digit++;

}

System.out.println("No.of characters are: " +chr);

System.out.println("No.of digits are: " +digit);

}

catch(Exception e)

{

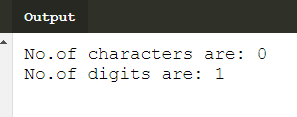
System.out.println(e);

}

}

}

Output:



1. **Inheritance** can be defined as the process where one class acquires the properties of another.The class which **inherits** the properties of other is known as subclass and the class whose properties are **inherited** is known as superclass .
2. **Multiple Inheritance** is a feature of object oriented concept, where a class can **inherit** properties of more than one parent class. The problem occurs when there exist methods with same signature in both the super classes and subclass.
3. 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.
4. A class that is declared using abstract keyword is known as abstract class. It can have abstract methods as well as concrete methods. An **abstract class** can not be instantiated, which means you are not allowed to create an object of it.
5. 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.
6. 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.
7. An interface is a completely "**abstract class**" that is used to group related methods with empty bodies. Syntax: interface<interface-name>{

// declare constant fields

//declare methods that abstract.

}

1. **Package** in [Java](https://www.geeksforgeeks.org/java/) is a mechanism to encapsulate a group of classes, sub packages and interfaces.
2. An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions. When an error occurs within a method, the method creates an object and hands it off to the runtime system.This block of code is called an exception handler.
3. **J**ava 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.