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.

class Publication{

String title;

int price;

}

class Book extends Publication{

int pg;

public Book(String t,int p,int pg){

super.title=t;

super.price=p;

this.pg=pg;

}

void display(){

System.out.println("This is Book class");

System.out.println("Title: "+super.title);

System.out.println("Price: Rs."+super.price);

System.out.println("Pages: "+pg);

}

}

class CD extends Publication{

int s;

public CD(String t,int p,int s){

super.title=t;

super.price=p;

this.s=s;

}

void display(){

System.out.println("This is CD class");

System.out.println("Title: "+super.title);

System.out.println("Price: Rs."+super.price);

System.out.println("Size: "+s+" MB");

}

}

class Shop{

public static void main(String[] args) {

Book b=new Book("Java",40,2);

CD c=new CD("CA",25,650);

b.display();

c.display();

}

}

Output :

This is Book class

Title: Java

Price: Rs.40

Pages: 2

This is CD class

Title: CA

Price: Rs.25

Size: 650 MB

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

class Base{

void show(){

System.out.println("This is the method in super class.");

}

}

class Derived extends Base{

void show(){

super.show();

System.out.println("This is the method in subclass.");

}

}

class Override{

public static void main(String args[]) {

Derived d = new Derived();

d.show();

}

}

Output :

This is the method in super class.

This is the method in subclass.

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

interface Shape {

void CalculateArea();

}

class Square implements Shape {

int a;

public Square(int a){

this.a=a;

}

public void CalculateArea() {

System.out.println("The area of square is: " + (a \* a));

}

}

class Circle implements Shape {

int a;

public Circle(int a){

this.a=a;

}

public void CalculateArea() {

System.out.println("The area of circle is: " + (3.14 \* a \* a));

}

}

class Triangle implements Shape {

int a, b;

public Triangle(int a,int b) {

this.a = a;

this.b = b;

}

public void CalculateArea() {

System.out.println("The area of triangle is: " + (0.5 \* a \* b));

}

}

public class Area {

public static void main(String[] args) {

Square s = new Square(4);

Triangle t = new Triangle(4,3);

Circle c = new Circle(10);

s.CalculateArea();

t.CalculateArea();

c.CalculateArea();

}

}

Output :

The area of square is: 16

The area of triangle is: 6.0

The area of circle is: 314.0

1. 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.

A.java:

package p1;

public class A{

public static void display(){

System.out.println("This is a display method in p1 package class A");

}

}

javac –d . A.java

B.java:

package p2;

import p1.A;

public class B{

public static void main(String args[]){

A.display();

}

}

javac –d . B.java

java p2.B

Output:

This is a display method in p1 package class A

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

public class count{

public static void main(String args[]){

String s=args[0];

int nc=0,cc=0;

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

if(s.charAt(i)=='0' ||s.charAt(i)=='1' ||s.charAt(i)=='2' ||s.charAt(i)=='3' ||s.charAt(i)=='4' ||s.charAt(i)=='5' ||s.charAt(i)=='6' ||s.charAt(i)=='7' ||s.charAt(i)=='8' ||s.charAt(i)=='9'){

nc+=1;

}else{

cc+=1;

}

}

System.out.println("number count:"+nc);

System.out.println("character count:"+cc);

}

}

Output:

javac count.java

java count 32kjdfs23

number count:4

character count:5

1. What is Inheritance?

**Inheritance** is a mechanism in which one class acquires the property of another class. In Java, the class which aquires the properties is called as derived class(sub-class) and the other class is called as base class(super class). The keyword extends is used by derived class to inherit the features of base class.

1. What is Multiple Inheritance?

Multiple Inheritance is one of the types of Inheritance, in which one class aquires the properities of more than one class. In Multiple Inheritance, there is one derived class and 2 or more base classes. Java donot support Multiple Inheritance.

1. What is the use of Super keyword?

The**super** keyword in java is a reference variable that is used to refer parent class objects. Super keyword can be used for three purposes. They are:

* Used to refer immediate parent class instance variable.
* Used to invoke immediate parent class method.
* Used to invoke immediate parent class constructor.

1. What is abstract method?

A method which does not have implementation is known as an abstract method. Abstract keyword is used to declare a abstract method.

Abstract methodtype methodname(){}

1. What is abstract class?

A class with atleast one abstract method is called as Abstract class. An abstract class must be declared with an abstract keyword.

1. What is the use of final modifier?

The final modifier can be associated with methods, classes and variables. Once declared final

* A final class cannot be instantiated.
* A final method cannot be overridden.
* A final variable cannot be reassigned.

1. What is interface? Write the syntax interface.

Interface can be said as an abstract class which has all of its methods as abstract. An Interface will have all of it’s methods abstract by default. An Interface is declared by using keyword interface.

Syntax :

interface name{

void method name();

}

1. What is package?

A javapackage 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.

1. What is exception?

An exception is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disrupted and the program terminates abnormally.

1. What is the use of finally block?

A **finally block** contains all the crucial statements that must be executed whether exception occurs or not. The statements present in this block will always execute regardless of whether exception occurs.