1.

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 Main{

public static void main(String[] args) {

Book b=new Book("Java",10,4);

CD c=new CD("CA",25,570);

b.display();

c.display();

}

}

Output:

This is Book class

Title: Java

Price: Rs.10

Pages: 4

This is CD class

Title: CA

Price: Rs.25

Size: 570 MB

2.

import java.util.\*;

class A{

void show(){System.out.print("I'm in A");}

}

class Main extends A{

void show(){System.out.print("I'm in Main");}

public static void main(String[] args) {

Main m=new Main();

m.show();

}

}

Output:

I'm in Main

3.

import java.util.\*;

interface Shape{

void calculateArea();

}

class Square implements Shape{

public void calculateArea(){System.out.println("calculating square area");}

}

class Circle implements Shape{

public void calculateArea(){System.out.println("calculating circle area");}

}

class Triangle implements Shape{

public void calculateArea(){System.out.println("calculating triangle area");}

}

class Main{

void show(){System.out.print("I'm in Main");}

public static void main(String[] args) {

Triangle t=new Triangle();

Circle c=new Circle();

Square s=new Square();

s.calculateArea();

c.calculateArea();

t.calculateArea();

}

}

Output:

calculating square area

calculating circle area

calculating triangle area

4.

A.java

package p1;

public class A{

public void display(){System.out.println("in class A");}

}

B.java

package p2;

import p1.\*;

class B{

public static void main(String[] args) {

A a=new A();

a.display();

}

}

Output:

In class A

5.

class Main{

public static void main(String[] args) {

String s=args[0];

int c=0,n=0;

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

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

n+=1;

else

c+=1;

}

System.out.println("numbers :"+n+"\ncharacters :"+c);

}

}

Output:

Javac Main.java

java Main ajay1234k

numbers :4

characters :5

1.

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class. Hence, inheritance facilitates Reusability

2.

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. 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.

An abstract method is a method that is declared without an implementation (without braces, and followed by a semicolon)

syntax

public abstract class class name{

abstract void method name();

}

5.

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).

abstract class A{}

6.

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.

7.

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();

}

8.

A package is a namespace that organizes a set of related classes and interfaces.

9.

An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions. 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.