Right stroke training

Assignment-3

1.Implementation of super,this and method overriding

Program:

```
class Publication
{
  String title;
  int price;
  public Publication(String title,int price)
  {
   this.title=title;
   this.price=price;
  //System.out.println("In publication constructor");
 }
  void display()
 {
  System.out.println("In publication,display method");
  System.out.println("Title: "+title+" Price: "+price);
 }
}
class Book extends Publication
 {
  int pages;
  Book(String title,int price,int pages)
  {
    super(title,price);
    this.pages=pages;
   }
   void display()
   {
    super.display();
    System.out.println("In book class");
    System.out.println("Title: "+title+" Price: "+price+" Pages: 
"+pages);
   }
}
class CD extends Book
{
  int size;
  CD(String title,int price,int pages,int size)
  {
    super(title,price,pages);
    this.size=size;
   }
    void display()
    {
     super.display();
     System.out.println("In CD class");
     System.out.println("Title: "+title+" Price: "+price+" Pages: 
"+pages+" Size: "+size);
    }
   }
   public class InheritResult
   {
    public static void main(String args[])
   {
    String title=args[0];
    int price=Integer.parseInt(args[1]);
    int pages=Integer.parseInt(args[1]);
    int size=Integer.parseInt(args[1]);
    CD cd=new CD(title,price,pages,size);
    cd.display();
   }
}
```

Output:

```
E:\fullstackjava>javac InheritResult.java
E:\fullstackjava>java InheritResult "Amazing life" 2400 5000 30
In publication,display method
Title: Amazing life Price: 2400
In book class
Title: Amazing life Price: 2400 Pages: 2400
In CD class
Title: Amazing life Price: 2400 Pages: 2400 Size: 2400
```

2.Method overriding

Program:

```
import java.lang.Math;
class Circle
{
public void area(int radius)
{
double areacircle=(double)Math.PI*radius*radius;
System.out.println("area of circle is"+areacircle);
}
}
class Semicircle extends Circle
{
static int radius=5;
public void area(int radius)
{
super.area(radius);
double areasemi=((double)Math.PI*radius*radius)/2;
System.out.println("area of semi circle is"+areasemi);
}
public static void main(String args[])
{
Semicircle sc=new Semicircle();
sc.area(radius);
}
}
```

Output:

```
E:\fullstackjava>javac Semicircle.java
E:\fullstackjava>java Semicircle
area of circle is78.53981633974483
area of semi circle is39.269908169872416
```

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

Ans:

```
import java.lang.Math;
interface Shape
{
   void CalculateArea();
}
class Square implements Shape
 {
   public void CalculateArea()
  {
   int side=4;
   int area=side*side;
   System.out.println("the area of square is "+area);
  }
}
class Circle implements Shape
{
  double radius=3;
  public void CalculateArea()
  {
  double area=Math.PI*radius*radius;
  System.out.println("the area of circle is "+area);
 }
}
class Triangle implements Shape
{
  public void CalculateArea()
 {
  int base=3,height=4;
  double area=0.5*base*height;
  System.out.println("the area of triangle is "+area);
 }
}
public class Test
{
public static void main(String args[])
 {
  Shape sh1=new Square();
  Shape sh2=new Circle();
  Shape sh3=new Triangle();
  sh1.CalculateArea();
  sh2.CalculateArea();
  sh3.CalculateArea();
 }
}
```

Output:

```
E:\fullstackjava>javac Test.java
E:\fullstackjava>java Test
the area of square is 16
the area of circle is 28.274333882308138
the area of triangle is 6.0
```

4.Package

Program:

A.java:

```
package p1;
public class A
{
```

```
public void display()
{
System.out.println("I am in package p1");
}
}
```

B.java:

```
package p2;
import p1.A;
class B
{
public static void main(String args[])
{
A a=new A();
a.display();
}
}
```

Output:

```
E:\fullstackjava>javac A.java
E:\fullstackjava>java A
Error: Could not find or load main class A
E:\fullstackjava>javac B.java
E:\fullstackjava>java B
I am in package p1
```

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

Program:

```
import java.util.*;
class Count
{
public static void main(String args[])
{
```

```
String s=args[0];
try
{
int digit=0;
int character=0;
for(int i=0;i<s.length();i++)
{
if(Character.isLetter(s.charAt(i)))
  character++;
else
  digit++;
}
System.out.println("no.of characters are: "+character+"\nno of digits
are: "+digit);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

Output:

```
E:\fullstackjava>javac Count.java
E:\fullstackjava>java Count Ab567Hgj89
no.of characters are: 5
no of digits are: 5
```

1.What is Inheritance?

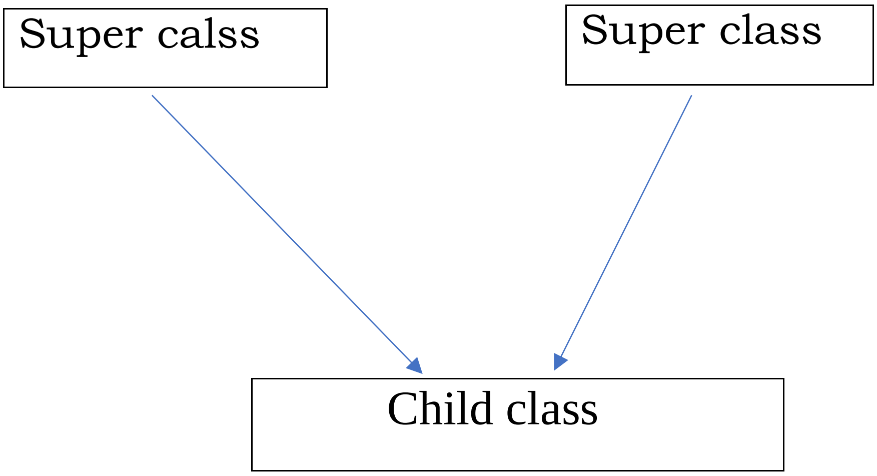
Ans: Inheritance is part of Object Oriented Programming. It is defined as the process of acquiring properties from one class to another class.so that we can manage the data

in hierarchical. The use of inheritance is we can create new class that are built based up on existing class.
In inheritance we have two classes. The class which provides properties to another class is called parent/base class and the class which access/acquires the properties from another class is called child/derived class.

2.What is Multiple Inheritance?

Ans:It is_one of the type of inheritance that allows to access the properties of one class to another class.
Multiple inheritance allows to derive one base class using one or more super classes.

One class can have more than one superclass and it access all properties from parent classes.
In java,multiple inheritance is not possible because a class can extends only one class.So,we use interface to achieve multiple inheritance.



3. What is the use of Super keyword?

Ans:Super is a special method which is used to invoke parent class variables,methods and constructors.

It is used in method overriding to access parent class method. It can invoke immediate parent class method. super() acts as immediate parent class constructor and should be the first line in child class constructor.

4.What is abstract method?

Ans: A method which is declared as abstract and does not have implementation is known as an abstract method.

Eg:

abstract void printStatus();//no method body and abstract.

5. What is abstract class?

Ans: A class which is declared as abstract is known as an **abstract class**. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.

- It can have constructors and static methods also.
- It can have final methods which will force the subclass not to change the body of the method.

6. What is the use of final modifier?

Ans:Final is a non-access modifier which can be used before method,variable and class.The method,variable and class which specified as final can’t be overridden,insitiated and reassigned Simply it can’t be changeable once declared.

7.What is interface? Write the syntax interface.

Ans:Interface is unimplemented structure.In interface we define only the methods but not the logic.A class takes responsibility to implement the methods if they implement that interface.

Unless the class that implements the interface is abstract, all the methods of the interface need to be defined in the class. Writing an interface is similar to writing a class. But a class describes the attributes and behaviors of an object. And an interface contains behaviors that a class implements.

Syntax:-

```
interface{

//methods
}
```

8.What is package?

Ans: Package in Java is a mechanism to encapsulate a group of classes, sub packages and interfaces. Packages are used to prevent naming conflicts,usage of classes, interfaces, enumerations and annotations easier.

Syntax:

```
package package_name;
```

9.What is exception?

Ans:Exception is a problem that arises during the execution of the program.Exception was meant to give you an opportunity to do something with it. like try something else or write to the log.Exceptions are raised by some user error or by programming errors.

These are handled by try,catch and finally block.

10. What is the use of finally block?

Ans:Finally is used in exception handling.It is used when a programmer wants to execute a piece of code even if the program has exceptions.Finally block guarantees that section of code will be executed.

It was written after try and catch block.The finally block always executes when the try block exits. But finally is useful for more than just exception handling,it allows the programmer to avoid having cleanup code accidentally bypassed by a return, continue, or break.