Assignment-3

1.Implementation of super,this and method overriding

Program:

```
import java.util.*;

class Publication

{

String title;

int price;

public Publication(String title,int price)

{

this.title=title;

this.price=price;

}

void display()

{
```

```
System.out.println("The price of the book "+title+" is "+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("The number of pages in the book is"+pages);
}
public static void main(String args[])
```

```
{
Scanner sc=new Scanner(System.in);
String title=sc.next();
int price=sc.nextInt();
int pages=sc.nextInt();
Book book=new Book(title,price,pages);
book.display();
}
}
```

input:

```
Sun
120
1200
```

output:

```
The price of the book sun is 120
The number of pages in the book is1200
```

2.write a program for Method overriding

Program:

```
class A
{
public void main(int a,int b)
{
System.out.println(“The sum of two numbers is “+ a+b);
}
}
class B extends A
{
public void main(int a,int b)
{
System.out.println(“The product of two numbers is “+ a*b);
}
public static void main(String args[])

{

B b=new B();

b.main(2,3);

}

}
```

Output:

```
The sum of two numbers is 5

The product of two numbers is 6
```

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

Program:-

```
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 the 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 the 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:

```
the area of square is 16

the area of circle is 28.274333882308138

the area of triangle is 6.0
```

4.Package

Program:

Package1.java:

```
package p1;

public class Package1

{
```

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

Package2.java:

```
package p2;
import p1.Package1;
class Package2
{
public static void main(String args[])
{
Package1 p=new Package1();
p.display();
}
}
```

Output:

```
E:\javafullstackassignments;javac Package2.java

E:\javafullstackassignments;java Package2

I am in package p1
```

5. Write a java program to count numbers, characters in the command line argument 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&lt;s.length();i++)
{
if(Character.isLetter(s.charAt(i)))
character++;
else
digit++;
}

System.out.println(“The no.of characters are ”+character);
System.out.println(“The no.of digits are ”+digit);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

Output:

```
 i/p:abc123

no.of characters are: 3

no of digits are: 3
```

1. What is Inheritance?

Ans:- 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) whose properties are inherited is known as superclass (base class, parent class).extends is the keyword used to inherit the properties of a class.

2. What is Multiple Inheritance?

Ans:- In Multiple inheritance, one class can have more than one superclass and inherit features from all parent classes.Java does not support Multiple inheritance.But we can implement multiple inheritance in java by using interface.

3. What is the use of Super keyword?

Ans:- The super keyword refers to superclass (parent) objects. It is used to call superclass methods, and to access the superclass constructor.

Use of super keyword

- super variable refers immediate parent class instance.
- super variable 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:- 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. What is interface? Write the syntax interface.

Ans: - An interface is a reference type in Java. It is similar to class. It is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface.

Along with abstract methods, an interface may also contain constants, default methods, static methods, and nested types. Method bodies exist only for default methods and static methods.

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.

Unless the class that implements the interface is abstract, all the methods of the interface need to be defined in the class.

Syntax:-interface variable{

//methods

}