ASSIGNMNET – 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

Program :

```
class Publication

{

String title;

int price;

Publication(String title,int price)

{

this.title=title;

this.price=price;

}

void display()

{

System.out.println("The title of the book is "+this.title+" and price is "+this.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("It consists of "+this.pages+" pages");
}
}
class CD extends Publication
{
int size;
CD(String title,int price,int size)
{
super(title,price);
this.size=size;
}
void display()
{
super.display();
System.out.println("It's size is "+this.size);
}
}
class imp
```

```
{

public static void main(String [] x)

{

Book obj = new Book("Harry Potter and the Philosopher's Stone",1000,5000);

obj.display();

CD ob = new CD("The Lord of the Rings",1500,500);

ob.display();

}

}
```

Output :

```
The title of the book is Harry Potter and the Philosopher's Stone and price is 1000

It consists of 5000 pages

The title of the book is The Lord of the Rings and price is 1500

It's size is 500
```

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

Program :

```
class method

{

void methodover()

{

System.out.println("This function is called from derived class");

}

}

class methodoverriding extends method

{

void methodover()
```

```
{

super.methodover();

System.out.println("Now we just called the base class function of the same name methodover using super keyword .\nThis is called method overriding ");

}

}

class overriding

{

public static void main(String [] x)

{

methodoverriding obj = new methodoverriding();

obj.methodover();

}

}
```

Output :

```
This function is called from derived class

Now we just called the base class function of the same name methodover using super keyword .

This is called 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.

Program :

```
class Square implements Shape

{

int side;

Square(int side)

{
```

```
this.side=side;
}
public void  CalculateArea()
{
System.out.println("The area of the square is "+(this.side*this.side));
}
}
class Circle implements Shape
{
double radius;
Circle(double radius)
{
this.radius=radius;
}
public void CalculateArea()
{
System.out.println("The area of the Circle is "+(3.147*(this.radius*this.radius)));
}
}
class Triangle implements Shape
{
double b,h;
Triangle(double b,double h)
{
this.b=b;
```

```
this.h=h;

}

public void CalculateArea()

{

System.out.println("The area of the triangle is "+(0.5*(this.b)*(this.h)));

}

}

class interfaceimp

{

public static void main(String [] x)

{

Square ob = new Square(5);

ob.CalculateArea();

Circle ob1 = new Circle(2.0);

ob1.CalculateArea();

Triangle ob2  = new Triangle(3,10);

ob2.CalculateArea();

}

}
```

Output :

```
The area of the square is 25

The area of the Circle is 12.588

The area of the triangle is 15.0
```

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.

Program :

```
/*package implementation of p1 */

package p1;

public class A

{

public void display()

{

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

}

}

/*Package implementation of p2*/

package p2;

import p1.*;

public class B

{

public static void main(String [] x)

{

p1.A obj = new p1.A();

}

public void disp()

{

p1.A obj = new p1.A();

obj.display();
```

```
}

}

/*using packages p1 and p2 */

import p2.*;

class AB

{

public static void main(String [] x)

{

p2.B ob = new p2.B();

ob.disp();

}

}
```

Output :

This is class A in package p1.

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

Program :

```
class Exceptions

{

public static void main(String [] x)

{

int c1=0,c2=0,i=0;

try

{

while (true){

char ch = x[i].charAt(0);
```

```
if(Character.isDigit(ch))
{
c1++;
}
else
{
c2++;
}
i++;
}
}
catch(Exception ie){}
finally
{
System.out.println("The number of digits entered is "+c1);
System.out.println("The number of characters entered is "+c2);
}
}
}
```

Output :

```
java Exceptions 1 2 3 a d s  4 r
The number of digits entered is 4
The number of characters entered is 4
```

Theoretical Questions:

1. What is Inheritance?

Inheritance: It is the process where one class derives the properties of another class. It is useful for code reusability.

2. What is Multiple Inheritance ?

Multiple Inheritance: Multiple Inheritance is a feature of object oriented concept, where a class can inherit properties of more than one parent class.

3. What is the use of Super keyword?

If the method overrides one of its super class's methods, overridden method can be invoked through the use of the keyword super. It can be also used to refer to a hidden field.

4. What is abstract method?

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.

5. What is abstract class?

These classes cannot be instantiated and are either partially implemented or not at all implemented. This class contains one or more abstract methods which are simply method declarations without a body.

6. What is the use of final modifier?

It is used to make a method or variable in order to make them constant so they won't be changed in further implementation.

7. What is interface? Write the syntax interface.

An interface is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface.

Syntax:

Interface interface name

8. What is package?

A package is collection of different classes and interfaces.

9. What is exception?

An exception is a problem that arises during the execution of a program. Exceptions are caught by handlers positioned along the thread's method invocation stack.

10. What is the use of finally block?

When an exception is occurred the remaining statements may not execute, such statements are to be specified in the finally block in order to execute.