**EX. NO: 7a INHERITANCE & PACKAGES**

**(15/3/2017-19/3/2017)**

**Note: Part I and Part 2 should be recorded in your observation as directed by your mentor**

**Part-I (Who should do Part-I?)**

***Anyone who wants to clear java, hope everyone wants to…….***

1. **Classes: Ship, CruiseShip**, and **CargoShip**

Design a **Ship** class that has the following members:

• A member variable for the name of the ship (a string)

• A member variable for the year that the ship was built (a string)

• A constructor and appropriate accessors and mutators

• A print function that displays the ship’s name and the year it was built.

Design a **CruiseShip** class that is derived from the Ship class.

The CruiseShip class should have the following members:

• A member variable for the maximum number of passengers (an int)

• A constructor and appropriate accessors and mutators

• A print function that overrides the print function in the base class. The

CruiseShip class’s print function should display only the ship’s name and the

maximum number of passengers.

Design a **CargoShip** class that is derived from the Ship class.

The CargoShip class should have the following members:

• A member variable for the cargo capacity in tonnage (an int).

• A constructor and appropriate accessors and mutators.

• A print function that overrides the print function in the base class. The

CargoShip class’s print function should display only the ship’s name and the

ship’s cargo capacity.

Demonstrate the classes in a program that has a Ship reference. The Ship reference should be initialized with the addresses of dynamically allocated Ship, CruiseShip, and CargoShip objects. The program should then call each object’s print function. [Hint: Dynamic method dispatch]

1. Define a package named **readData** with the following Classes:

**StringValue** - contains readString() method, which can read a string

**IntegerValue** - contains readInteger() method, which can read an integer

**DoubleValue** - contains readDouble() method, which can read an double

Define a class **readDataDemo** which contains main method. Within the main() method, create objects for StringValue, IntegerValue and DoubleValue classes and invoke the readString(), readInteger() and readDouble() methods.

1. Create a base class called **Vehicle** that has the manufacturer’s name (type string) and number of cylinders in the engine (type int). Then create a class called **Truck** that is derived from Vehicle and has additional properties: the load capacity in tons (type double since it may contain a fractional part) and towing capacity in pounds (type int). Be sure your classes have a reasonable complement of constructors, accessor, and mutator methods. Write a driver program that tests all your methods.

Define a **Car** class that is derived from the Vehicle class and define a class called **SportsCar** that is derived from Car class. Be creative in choosing member variables and functions. Write a driver programs to test the Car and SportsCar classes.

1. Develop a standalone application in Java to demonstrate the concept of Finalize and Constructors in inheritance.

**Part-II: MCQ (Who should do Part-II?)**

***Who will not love fun? If you love fun go ahead …and you can take this fun back home and continue there as well……………….***

1. Consider the following program:

class Base {

public static void foo(Base bObj) {

System.out.println("In Base.foo()");

bObj.bar();

}

public void bar() {

System.out.println("In Base.bar()");

}

}

class Derived extends Base {

public static void foo(Base bObj) {

System.out.println("In Derived.foo()");

bObj.bar();

}

public void bar() {

System.out.println("In Derived.bar()");

}

}

class OverrideTest {

public static void main(String []args) {

Base bObj = new Derived();

bObj.foo(bObj);

}

}

What is the output of this program when executed?

a)

In Base.foo()

In Base.bar()

b)

In Base.foo()

In Derived.bar()

c)

In Derived.foo()

In Base.bar()

d)

In Derived.foo()

In Derived.bar()

**Part-III (Who should do Part-III?)**

***If you are a person who loves to challenge yourself, train yourself till you tire and in short for those who aspire to become extra intellect, this is for you***

1. Extend or Modify your Bank Application to incorporate the appropriate use Inheritance and Packages.