**Lab Assignment 1**

Student Id:30340670

Course Code:ITECH 2306

Date: 14-05-020

1. **Name a super class and describe its purpose.**

Answer: Property.java is a super class, in this class, basic requirement covered and functionality also for all class further implementation, according to their implement child class used basic things from super class or parent class as well as its specific functionality

1. **Name a sub class and describe what inheritance means for that class.**

Answer: Sub class Name are: Residential, Commercial, Hospital, Industrial, School, VacantLand and Other.

1. **Name a method that is overridden in two sub classes.**

Answer: toString() method.

1. **Identify where a super constructor is referred to in the code.**

Answer: In the child class constructor, in first line call super constructor.

Example:

public abstract class Property {

public Property()

{}

}

public class Residential {

public Residential()

{ super(); }

}

1. **Explain the classes and methods involved in a calculation of extra services on a property type of your choice.**

Answer: In the Driver class, there are many other for calculation and manager extra service on a specific property.

1. **Overloading of a method or constructor**

Answer: In overloading of a method, in which a class having method name is same but different argument types, it is same as with constructor but constructor have no return type.

Example:   
public abstract class Property {

private String alotNumber;

public Property()

{}

public Property(String alotNumber)

{ this.alotNumber = alotNumber; }

}

1. **Overriding of concrete methods in sub classes**

Answer: In Overriding of concrete method, in which method signature should be same, and same number of argument and data type also as same as the parent class, then only override the concrete methods, In other words, not happy with parent class logic implement in the concrete, then child class inherit from parent and not satisfied with parent class implementation, then only override the method and provide different logic implementation. Example:

public abstract class Property {

private String alotNumber;

public int testing(int a, int b)

{ return a+b; }

}

public class Residential extends Property

{

private String type;

public int testing(int a, int b)

{ return (a\*b); }

}

1. **Abstract class**

Answer: A class having abstract keyword, and it may include abstract method or non-abstract method. Abstract class object can’t be instantiated, but it is possible with its sub classes. Example:

public abstract class Property {

private String alotNumber;

public Property(String alotNumber)

{ this.alotNumber = alotNumber;}

}

public class Residential extends Property

{

private String type;

public Residential(String alotNumber, String type)

{

super(alotNumber);

this.type = type;

}

}

1. **Abstract method and overriding of an abstract method**

Answer: In Abstract method, in which if any method contains abstract keyword in method signature, and its class also abstract then called as abstract method. By Child class implement the abstract class and provide implementation of the abstract method, if child class also not provide implementation for abstract method then its child class make as abstract and provide further implementation in its child class.

Example:

public abstract class Property {

public abstract void calculateValue();

}

public class Residential extends Property{

public void calculateValue(){

//logic define…

// …..

}

}

1. **Information Hiding and Encapsulation**

Answer: In information hiding and encapsulation, in which wrapping of a data in a single class unit. No outer class can access data directly, by using getter and setter methods to make visible data variable for outer class to access it. By using private keyword, we can achieve this.

Example:

public abstract class Property {

private String lotNumber;

private String statetGovernmentRegistrationIndex;

public String getLotNumber() {

return lotNumber;

}

public void setLotNumber(String lotNumber) {

this.lotNumber = lotNumber;

}

public String getStatetGovernmentRegistrationIndex() {

return statetGovernmentRegistrationIndex;

}

public void setStatetGovernmentRegistrationIndex(String statetGovernmentRegistrationIndex) {

this.statetGovernmentRegistrationIndex= statetGovernmentRegistrationIndex;

}

}