**Object Oriented Programming with Java**

**Lab Practice:5**

1. Consider the following code: 20

*public class A {*

*public void One(int i) {*

*}*

*public void Two(int i) {*

*}*

*public static void Three(int i) {*

*}*

*public static void Four(int i) {*

*}*

*}*

*public class B extends A {*

*public static void One(int i) {*

*}*

*public void Two(int i) {*

*}*

*public void Three(int i) {*

*}*

*public static void Four(int i) {*

*}*

*}*

Answer the followings:

1. Which method overrides a method in the superclass?
2. Which method hides a method in the superclass?
3. What do the other methods do?

80

1. Create a class name ‘Person’. The class contains two fields; String calle firstName and lastName and the following methods.
   1. default and alternate constructors in the class.
   2. two accessors (getter) to return the first and the last name.
   3. A method called setName to set the fields to the parameters passed.
   4. A method called print (should print first and last)
   5. A method name toString()
   6. a method name equals (pass an object of the Object class)
   7. two methods name copy and getCopy to make a copy of the Person object into another Person object.

Note: the Person class is the super class for a class called employee. And this employee class should contain three fields (payRate, workHours, and deptName)

Consider the following structure of code.

*//Class Employee: subclass of Person*

*public class Employee extends Person {*

*private double payRate;*

*private double workHours;*

*private String deptName;*

*public final int HOURS = 35;*

*public final double OVERTIME = 1.2;*

*//default constructor*

*public Employee() {*

*...*

*}*

*//add an alternate constructor with parameters*

*public String toString() {*

*//should return a String like this:*

*//The wages for xxxx from the xxxx department are: $xxxxx.xx"*

*...*

*}*

*public void print() {*

*//Should print output like this (same line):*

*//The employee xxxx from the xxxx department worked xx hours*

*//with a pay rate of $xxx.xx. The wages for this employee are $xxxxx.xx*

*...*

*}*

*public double calculatePay() {*

*//Method to calculate and return the wages*

*//handle both regular and overtime pay*

*...*

*}*

*public void setAll(String first, String last, double rate, double hours, String dep){*

*...*

*}*

*public double getPayRate() {*

*…*

*}*

*public double getHoursWorked() {*

*…*

*}*

*public String getDepartment() {*

*...*

*}*

*public boolean equals(Object o) {*

*...*

*}*

*public Employee getCopy() {*

*...*

*}*

*public void copy(Employee e) {*

*...*

*}*

*}}*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*additional Questions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Topic: Defining Class Relationship

3. Create a class “Programmer” that is a subclass of class “Employee” in question 2. The relationship between the two classes is Programmer IS-A Employee. Then, override proper method so that the programmer object gets 10% of the salary as a bonus as a bonus.

Finally, let the “Dog” class below be a "HAS-A" relationship between the Programmer class and the Dog class. Then, create and call a "feed" method where the programmer object calls the dog class’s eat method.

public class Dog{

private String name;

private int age;

public void eat( ){

System.out.println(“my dog is eating!”);

}

}