ITC142: Homework 4

Amrit Dahal

Seattle Central College Sp2019

1. Define the Reptile class as a subclass of the Pet class. The speak method

returns an empty string.

→ class Reptile extends Pet

{

Public String speak()

{

Return “”;

}

}

2. Which one of the following statements is valid?

Pet p = new Cat();

Cat c = new Pet();

→ a

3. Is the following code valid?

Pet p = new Dog();

System.out.println(p.fetch( ));

→ Not Valid- must be typecast as (Dog)p.fetch().

4. Which is the subclass and which is the superclass in this declaration?

class X extends Y { ... }

→ Subclass: X

Superclass: Y

5. Which visibility modifier allows the data members of a superclass to be

accessible to the instances of subclasses?

→ protected

6. Suppose Truck and Motorcycle are subclasses of Vehicle. Which of these

declarations are invalid?

Truck t = new Vehicle();

Vehicle v = new Truck();

Motorcycle m1 = new Vehicle();

Motorcycle m2 = new Truck();

→ a, c, d

7. What is the purpose of the instanceof operator?

→ Used to check the type of object; determine if object is instance of specified class or not.

8. If X is a private member of the Super class, is X accessible from a subclass of

Super?

→ No, private cannot be read by anything outside the class.

9. If X is a protected member of the Super class, is X of one instance accessible

from another instance of Super? What about from the instances of a subclass

of Super?

→ Yes. Yes- protected and public.

10. How do you call the superclass’s constructor from its subclass?

→ using the super(), and passing correct number and types for parameters, as first statement of subclass constructor.

11. What statement will be added to a constructor of a subclass if it is not included

in the constructor explicitly by the programmer?

→ reserved word *super()* with no arguments.

12. Modify the definition of GraduateStudent and UndergraduateStudent in

Section 13.1 so we can create their instances in this way:

student1 = new UndergraduateStudent();

student2 = new UndergraduateStudent("Mr. Espresso");

student3 = new GraduateStudent();

student4 = new GraduateStudent("Ms. Latte");

→   
 class UndergraduateStudent extends Student {

Public UndergraduateStudent(string undergradName) {

super(undergradName);

}

Public UndergraduateStudent() {

super();

}

}

class GraduateStudent extends Student {

Public GraduateStudent(string gradName) {

super(gradName);

}

Public GraduateStudent() {

super();

}

}

13. Can you create an instance of an abstract class?

→ No.

14. Must an abstract class include an abstract method?

→ No.It can inherit abstract methods from superclass.

15. What is wrong with the following declaration?

class Vehicle {

abstract public getVIN();

...

}

→ Vehicle{} must be declared abstract as it contains abstract member.