ITMD 411 Week 2-3 Review Chapter 6 OOP

1) Methods that operate on an object's fields are called

A) Private methods **B) Instance methods** C) Instance variables D) Public methods

2) A constructor

A) Has return type of void B) Always accepts two arguments

C) Always has an access specifier of private **D) Has the same name as the class**

3) A method that gets a value from a class's field but does not change it is known as a mutator method. A) True **B) False**

4) Given the code below, what is the value of finalAmount when it is displayed?

public class Order {

private int orderNum;

private double orderAmount;

private double orderDiscount;

public Order(int orderNumber, double orderAmt,double orderDisc) {

orderNum = orderNumber;

orderAmount = orderAmt;

orderDiscount = orderDisc;

}

public int getOrderAmount() {

return orderAmount;

}

public int getOrderDisc() {

return orderDiscount;

}

}

public class CustomerOrder {

public static void main(String[] args) {

int ordNum = 1234;

double ordAmount = 580.00;

double discountPer = .1;

Order order;

double finalAmount = order.getOrderAmount() -

order.getOrderAmount() \* order.getOrderDisc();

System.out.println("Final order amount = $" +

finalAmount);

}

}

A) 580.00 **B) There is no value because the object order has not been created**

C) 528.00 D) There is no value because the constructor has an error

5) In UML diagrams, this symbol indicates that a member is private.

A) # **B) -** C) + D) \*

Notes on UML base relationships.

**Dependency Diagram**

Defined as the relationship between two classes means such that class **A** depends on class **B**, or that A uses B.

**A**

**B**

Given this situation,thisunidirectional dependency relationship is restricted to the following meaning.

a. Class **A** receives an instance of class **B** as a parameter to at least one of its methods, or

b. Class **A** creates an instance of class **B**, local to one of its methods.

**Association Diagram**

The strength of an association relationship class diagram means that class **A** will contain at least one *instance* variable of class **B**, which makes class **B** structurally a part of class **A**.

Example:

**Name**

-firstName: String

-lastName: String

+Name(first: String, last: String)

+getFirstName(): String

+getLastName(): String

**Person**

**-name: Name**

-phone: String

+Person(n: Name, phone: String)

+getName(): Name

+getPhone(): String

Here Person is a composition of two fields, the first as the type Name, the second type String. This relationship establishes the fact that the class Name forms part of the physical structure of the class Person. Hence if we were to remove the field name from the class Person, and having just the field, phone, we would not establish that we are talking about person. Hence by including the field, name to Person, at least intuitively adds the meaning person.

6. Given the scenario below what kind of relationship exists? **Dependency** or Association?

public class A

{

public void method1(B b)

{

}

public void method2()

{

B b = new B();

}

}

}

public class B

{

public void method()

{

}

public void method2()

{

}

}

1. Given that Q represents a class. Which of the following classes establish a dependency relationship of class P on class Q?

(c) class P {

P( ) { }

}

}

(b) class P {

P( ) {

Q q = new Q( );

}

}

}

**(a) class P {**

**Q q;**

**P( ) { q = new Q( ); }**

**}**

}