Quiz 2

**QUESTION 1**

A constructor

|  |  |  |
| --- | --- | --- |
|  |  | Has return type of void |
|  |  | Always has an access specifier of private |
|  |  | Always accepts two arguments |
|  |  | Has the same name as the class |

**QUESTION 2**

Given the following code, what will be 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 double finalOrderTotal()   
 {   
 return orderAmount - orderAmount \*  
 orderDiscount;  
 }   
}    
  
public class CustomerOrder    
{    
 public static void main(String[] args)   
 {   
 Order order;  
 int orderNumber = 1234;  
 double orderAmt = 580.00;  
 double orderDisc = .1;  
 order = new Order(orderNumber, orderAmt, orderDisc);  
 double finalAmount = order.finalOrderTotal();  
 System.out.println("Final order amount = $" +  
  finalAmount);  
 }   
}

|  |  |  |
| --- | --- | --- |
|  |  | 580.00 |
|  |  | 528.00 |
|  |  | 522.00 |
|  |  | There is no value because the object order has not been created. |

**QUESTION 3**

Look at the following code. Which line has an error?  
  
Line 1 public interface Interface1  
Line 2 {  
Line 3 int FIELDA = 55;  
Line 4 public int methodA(double){}  
Line 5 }

|  |  |  |
| --- | --- | --- |
|  |  | 1 |
|  |  | 2 |
|  |  | 3 |
|  |  | 4 |
|  |  |  |

**QUESTION 4**

Look at the following code.  
  
Line 1 public class ClassA  
Line 2 {  
Line 3 public ClassA() {}  
Line 4 public void method1(){}  
Line 5 }  
Line 6 public class ClassB extends ClassA  
Line 7 {  
Line 8 public ClassB(){}  
Line 9 public void method1(){}  
Line 10 }  
Line 11 public class ClassC extends ClassB  
Line 12 {  
Line 13 public ClassC(){}  
Line 14 public void method1(){}  
Line 15 }  
  
Which method1 will be executed as a result of the following statements?  
  
ClassA item1 = new ClassC();  
item1.method1();

|  |  |  |
| --- | --- | --- |
|  |  | Line 9 |
|  |  | Line 4 |
|  |  | Line 14 |
|  |  | This is an error and will cause the program to crash |

**QUESTION 5**

The generic method   
  
 public static <E extends Number>   
 void displayArray(E[] array)  
 {  
 for (E element : array)  
 System.out.println(element);  
 }  
  
can be passed

|  |  |  |
| --- | --- | --- |
|  |  | an array whose element type is Integer |
|  |  | an array whose element type is any superclass of Number |
|  |  | an array whose element type is Object |
|  |  | an array whose element type is E |

**QUESTION 6**

Which of the following statements declares Salaried as a subclass of PayType?

|  |  |  |
| --- | --- | --- |
|  |  | public class Salaried derivedFrom(Paytype) |
|  |  | public class PayType derives Salaried |
|  |  | public class Salaried extends PayType |
|  |  | public class Salaried implements PayType |
|  |  |  |

**QUESTION 7**

In UML diagrams, this symbol indicates that a member is public.

|  |  |  |
| --- | --- | --- |
|  |  | \* |
|  |  | # |
|  |  | - |
|  |  | + |

**QUESTION 8**

The following statement creates an ArrayList object. What is the purpose of the <String> notation?  
  
ArrayList<String> arr = new ArrayList<String>();

|  |  |  |
| --- | --- | --- |
|  |  | It specifies that only String objects may be stored in the ArrayList object |
|  |  | It specifies that the get method will return only String objects |
|  |  | It specifies that String objects may *not* be stored in the ArrayList object |
|  |  | It specifies that everything stored in the ArrayList object will be converted to a String |

**QUESTION 9**

You should not define a class field that is dependent upon the values of other class fields

|  |  |  |
| --- | --- | --- |
|  |  | In order to avoid having stale data |
|  |  | Because it is redundant |
|  |  | Because it should be defined in another class |
|  |  | In order to keep it current |

**QUESTION 10**

Instance methods should be declared static.

 True

 False