**Sample Quiz 2:**

1. **For the following code, which statement is *not* true?   
   public class Sphere   
   {   
     private double radius;   
     public double x;   
     private double y;   
     private double z;   
   }**

|  |  |  |
| --- | --- | --- |
|  |  | z is available to code that is written outside the Sphere class |
|  |  | radius is not available to code written outside the Sphere class |
|  |  | radius, x, y, and z are called members of the Sphere class |
|  |  | x is available to code that is written outside the Sphere class |

1. 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 int getOrderAmount()  
     {  
     return orderAmount;  
     }  
     public int getOrderDisc()  
     {  
     return orderDisc;  
     }  
   }

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);  
  }   
}

|  |  |  |
| --- | --- | --- |
|  |  | 528.00 |
|  |  | There is no value because the object order has not been created |
|  |  | 580.00 |
|  |  | There is no value because the constructor has an error |

1. **If ClassA is derived from ClassB, then**

|  |  |  |
| --- | --- | --- |
|  |  | Public and private members of ClassB are public and private, respectively, in ClassA |
|  |  | Neither public or private members in ClassB can be directly accessed in ClassA |
|  |  | Private members in ClassB are changed to protected members in ClassA |
|  |  | Public members in ClassB are public in ClassA, but private members in ClassB cannot be directly accessed in ClassA |

1. **Let Point<T> be a generic type. We want to write a method that takes as parameter Point objects whose type parameter is the Number class, or any subclass of Number. We can do this by declaring the type of the method parameter as**

|  |  |  |
| --- | --- | --- |
|  |  | Point<? super Number> |
|  |  | Point<? extends Number> |
|  |  | Point<? sub Number> |
|  |  | Point<Number> |

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

|  |  |  |
| --- | --- | --- |
|  |  | Public methods |
|  |  | Private methods |
|  |  | Instance variables |
|  |  | Instance methods |

1. **One of the advantages of using generics is**

|  |  |  |
| --- | --- | --- |
|  |  | that programs that use generics are smaller when translated to byte code |
|  |  | that more type problems can be uncovered at compile-time rather than at run time |
|  |  | that programs that use generic code require less effort in design and development |
|  |  | that program that use generics execute faster than programs that do not |

1. **The scope of a private instance field is**

|  |  |  |
| --- | --- | --- |
|  |  | Inside the class, but not inside any method |
|  |  | The instance methods of the same class |
|  |  | The method in which they are defined |
|  |  | Inside the parentheses of a method header |

1. **Two or more methods in a class may have the same name as long as**

|  |  |  |
| --- | --- | --- |
|  |  | You cannot have two methods with the same name |
|  |  | They have different return types, but the same parameter list |
|  |  | They have different return types |
|  |  | They have different parameter lists |

1. **What does the following UML diagram entry mean?**

**+ setHeight(h : double) : void**

|  |  |  |
| --- | --- | --- |
|  |  | This is a private field called Height and is a double data type |
|  |  | This is a public method with a parameter of data type double and does not return a value |
|  |  | This is a public field called Height and is a double data type |
|  |  | This is a private method with no parameters and returns a double data type |

1. **What is wrong with the following code?  
   public class ClassB extends ClassA  
   {  
     public ClassB()  
     {  
     int init = 10;  
     super(40);  
     }  
   }**

|  |  |  |
| --- | --- | --- |
|  |  | Nothing is wrong with the code |
|  |  | No values may be passed to super |
|  |  | The call to the method super must be the first statement in the constructor |
|  |  | The method super is not defined |