* **Question 1**

2 out of 2 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | A constructor  **Has the same name as the class** |  |  |  |
| |  | | --- | |  | |  |  |  |

* **Question 2**

2 out of 2 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | All fields declared in an interface  **Are final and static** |  |  |  |
| |  | | --- | |  | |  |  |  |

* **Question 3**

2 out of 2 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | Every class is either directly or indirectly derived from the Object class.  **True** |  |  |  |
| |  | | --- | |  | |  |  |  |

* **Question 4**

2 out of 2 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | 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);  }  }  **Final order amount = $522.00** |  |  |  |
| |  | | --- | |  | |  |  |  |

* **Question 5**

2 out of 2 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | 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 } |  |  |  |
| |  | | --- | | **Line 4** | |  |  |  |

* **Question 6**

2 out of 2 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | 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 14** |  |  |  |
| |  | | --- | |  | |  |  |  |

* **Question 7**

2 out of 2 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | Given the declaration:  ArrayList<Integer> array = new ArrayList<>(5);  what exception would be thrown with the following statement assuming the  arraylist has been fully populated up to the initialized size of 5 Integers:  int value = array.get(5);  **IndexOutofBoundsException** |  |  |  |
| |  | | --- | |  | |  |  |  |

* **Question 8**

2 out of 2 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | This is a group of related classes.  **package** |  |  |  |
| |  | | --- | |  | |  |  |  |

* **Question 9**

2 out of 2 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
|  | Which of the following statements declares Salaried as a subclass of PayType?  **public class Salaried extends PayType** |  |  |  |
| |  | | --- | |  | |  |  |  |

* **Question 10**

2 out of 2 points

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | | | | |
|  | There are two ways to overload methods in java, one of the ways is what?  **Use different arguments in the data types** | |  |  |  |
|  |  |  |  |  |