**QUESTION 1**

1. A search for an item X in a portion of a sorted array works by repeatedly selecting the middle item and comparing it to X. If X is not found there, the search method selects either the portion of the array to the left of the middle item, or the portion of the array to the right of the middle item, and continues the search. This method is called

|  |  |  |
| --- | --- | --- |
|  |  | binary search |
|  |  | sequential search |
|  |  | selection search |
|  |  | None of the above |

**0 points**

**QUESTION 2**

1. All fields declared in an interface

|  |  |  |
| --- | --- | --- |
|  |  | Have protected access |
|  |  | Are final and static |
|  |  | Must be initialized in the class implementing the interface |
|  |  | Have private access |

**0 points**

**QUESTION 3**

1. All of the exceptions that you will handle are instances of classes that extend this class.

|  |  |  |
| --- | --- | --- |
|  |  | RunTimeException |
|  |  | IOException |
|  |  | Exception |
|  |  | Error |

**0 points**

**QUESTION 4**

1. An exception's default error message can be retrieved using this method.

|  |  |  |
| --- | --- | --- |
|  |  | getDefaultErrorMessage() |
|  |  | getMessage() |
|  |  | getErrorMessage() |
|  |  | getDefaultMessage() |

**0 points**

**QUESTION 5**

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 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. |

**0 points**

**QUESTION 6**

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

**0 points**

**QUESTION 7**

1. If a class contains an abstract method,

|  |  |  |
| --- | --- | --- |
|  |  | The method will have only a header, but not a body, and end with a semicolon |
|  |  | The method must be overridden in subclasses |
|  |  | You cannot create an instance of the class |
|  |  | All of the above |

**0 points**

**QUESTION 8**

1. If ClassA is derived from ClassB, then

|  |  |  |
| --- | --- | --- |
|  |  | Public and private members of ClassB are public and private, respectively, in ClassA |
|  |  | Public members in ClassB are public in ClassA, but private members in ClassB cannot be directly accessed 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 |

**0 points**

**QUESTION 9**

1. Look at the following code:  
     
   FileInputStream fstream =   
     new FileInputStream("MyInfo.dat");  
   DataInputStream inputFile =   
     new DataInputStream(fstream);  
     
   This code can also be written as

|  |  |  |
| --- | --- | --- |
|  |  | DataInputStream inputFile =   new DataInputStream(new FileInputStream("MyInfo.dat")); |
|  |  | DataInputStream inputFile =    new DataInputStream("InputFile.txt"); |
|  |  | FileInputStream fstream =   new DataInputStream("InputFile.txt"); |
|  |  | FileInputStream inputFile =   new FileInputStream(new DataInputStream("MyInfo.dat")); |

**0 points**

**QUESTION 10**

1. Look at the following code. The method in line \_\_\_\_\_\_\_\_ will override the method in line \_\_\_\_\_\_\_\_.  
     
   Line 1 public class ClassA  
   Line 2 {  
   Line 3 public ClassA() {}  
   Line 4 public int method1(int a){}  
   Line 5 public int method2(int b){}  
   Line 6 }  
   Line 7 public ClassB extends ClassA  
   Line 8 {  
   Line 9 public ClassB(){}  
   Line 10 public int method1(int b){}  
   Line 11 public int method2(double c){}  
   Line 12 }

|  |  |  |
| --- | --- | --- |
|  |  | 5, 11 |
|  |  | 11, 5 |
|  |  | 4, 10 |
|  |  | 10, 4 |

**0 points**

**QUESTION 11**

1. 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 |

**0 points**

**QUESTION 12**

1. Look at the following code. Which line will cause a compiler error?  
     
   Line 1 public class ClassA  
   Line 2 {  
   Line 3 public ClassA() {}  
   Line 4 public final int method1(int a){}  
   Line 5 public double method2(int b){}  
   Line 6 }  
   Line 7 public ClassB extends ClassA  
   Line 8 {  
   Line 9 public ClassB(){}  
   Line 10 public int method1(int b){}  
   Line 11 public double method2(double c){}  
   Line 12 }

|  |  |  |
| --- | --- | --- |
|  |  | 5 |
|  |  | 10 |
|  |  | 4 |
|  |  | 11 |

**0 points**

**QUESTION 13 ???**

1. Look at the following code.  
     
   Integer myNumber = new Integer(5);  
   int var = myNumber;  
     
   Which of the following is true about the second statement?

|  |  |  |
| --- | --- | --- |
|  |  | The statement performs autoboxing. |
|  |  | The statement performs unboxing. |
|  |  | The statement performs unwrapping. |
|  |  | It results in an error because you cannot assign a wrapper class object to a primitive variable. |

**0 points**

**QUESTION 14**

1. One of the design tools used by programmers when creating a model of the program is

|  |  |  |
| --- | --- | --- |
|  |  | Compiler |
|  |  | Pseudocode |
|  |  | Disk drive |
|  |  | ALU |

**0 points**

**QUESTION 15**

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

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

**0 points**

**QUESTION 16**

1. 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 E |
|  |  | an array whose element type is Object |

**0 points**

**QUESTION 17**

1. What is the value of scores[2][3] in the following array?  
     
   int [] [] scores = { {88, 80, 79, 92}, {75, 84, 93, 80},  
     {98, 95, 92, 94}, {91, 84, 88, 96} };

|  |  |  |
| --- | --- | --- |
|  |  | 84 |
|  |  | 95 |
|  |  | 93 |
|  |  | 94 |

**0 points**

**QUESTION 18**

1. What will be displayed as a result of executing the following code?  
     
   int x = 6;  
   String msg = "I am enjoying this class.";  
   String msg1 = msg.toUpperCase();  
   String msg2 = msg.toLowerCase();  
   char ltr = msg.charAt(x);  
   int strSize = msg.length();  
   System.out.println(msg);  
   System.out.println(msg1);  
   System.out.println(msg2);  
   System.out.println("Character at index x = " +  
     ltr);  
   System.out.println("msg has " + strSize +  
     "characters.");

|  |  |  |
| --- | --- | --- |
|  |  | I am enjoying this class. I AM ENJOYING THIS CLASS. i am enjoying this class. Character at index x = e msg has 25 characters. |
|  |  | I am enjoying this class. I AM ENJOYING THIS CLASS. i am enjoying this class. Character at index x = n msg has 24 characters. |
|  |  | I am enjoying this class. I AM ENJOYING THIS CLASS. i am enjoying this class. Character at index x = e msg has 24 characters. |
|  |  | I am enjoying this class. I AM ENJOYING THIS CLASS. i am enjoying this class. Character at index x = n msg has 25 characters. |

**0 points**

**QUESTION 19**

1. What will be returned from the following method?  
     
   public static int methodA()  
   {  
     double a = 8.5 + 9.5;  
     return a;  
   }

|  |  |  |
| --- | --- | --- |
|  |  | This is an error |
|  |  | 8 |
|  |  | 18 (as an integer) |
|  |  | 18.0 |

**0 points**

**QUESTION 20**

1. What will be the values of ans, x, and y after the following statements are executed?  
     
   int ans = 35, x = 50, y =50;  
   if ( x >= y)  
   {  
     ans = x + 10;  
     x -=y;  
   }  
   else  
   {  
     ans = y + 10;  
     y += x;  
   }

|  |  |  |
| --- | --- | --- |
|  |  | ans = 60, x = 50, y =100 |
|  |  | ans = 60, x = 0, y = 50 |
|  |  | ans = 45, x = 50, y = 0 |
|  |  | ans = 45, x = 50, y = 50 |