**Grading Summary**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | These are the automatically computed results of your exam. Grades for essay questions, and comments from your instructor, are in the "Details" section below. | |  |  | | --- | --- | | Date Taken: | **2/27/2014** | | Time Spent: | **1 h , 36 min , 16 secs** | | Points Received: | **38.98 / 40  (97.5%)** | | | |  |  |  | | --- | --- | --- | | Question Type: | # Of Questions: | # Correct: | | Short | 14 | N/A | | Essay | 6 | N/A | | | |

|  |
| --- |
| **Grade Details - All Questions** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Page:** | 1  [2](https://takeexam.next.ecollege.com/(NEXT(743c89e409))/Main/CourseMode/StudentGradebookExam/StudentGradebookExamView.ed?examID=53198966&courseItemSubId=444416170&studentID=22698822&currentPageNumber=2&digest=xqrAsOnUq9USxjt5Vmt6Ns9tamulNHbKUvHZ5TFasvI%253d&) |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 1. | Question : | Write the definition of an expression. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | An expression is a collection of variables and primitive data types that form an algorithm. | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0.5 of 0.5** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 2. | Question : | Write the definition of an algorithm. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | An algorithm is a set of instructions within a program that utilizes expressions, variables, and other primitive types in order to accomplish a task. This utilizes mathematics and syntax to accomplish the goals of the programmer. | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0.5 of 0.5** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 3. | Question : | Describe the purpose of a Java class constructor. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | A java constructor is the blueprint by which every class is initialized and contains the instructions that are accomplished upon every creation of an object. They contain the instructions that will initialize every object every time that a new object is created. This unifies the way that classes are represented and created within the program. | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0.5 of 0.5** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 4. | Question : | What is the purpose of set<attribute> and get<attribute> methods? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | The Get attribute allows the class to return private variables that exist within the class. The get attribute ensures that only the information within the instanced class are returned and that there is no interference between "global" and class attributes. The Set attribute allows the class to set private variables that exist within the class. The set attribute ensures that only the information within the instanced class are edited and that there is no interference between "global" and class attribute settings. | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0.5 of 0.5** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 5. | Question : | Provide a definition of the term ***syntax.*** | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Syntax is the rules that regulate a specific programming languages inputs.  It basically exists as the framework that each language's compiler "understands" the input from the programmer. These rules layout the design and function of the algorithms that are presented by the programmer to the compiler and thus turned into system "Byte" code. | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0.5 of 0.5** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 6. | Question : | Provide a definition of the term ***semantics.*** | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | semantics within computer programming are the mathematical rules that govern the way that each variable and primitive type are able to interact within the program. This information is the basis by which all formulas and mathematical variation is accomplished within the program. Syntax and mathematics form semantics and are the overall way that programs are able to accomplish their goals. | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0.5 of 0.5** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 7. | Question : | Explain the term ***instance variable.*** | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | An instance variable is a variable that can be edited outside of the class by the instance.  They are referred to as the private variables within the class that can only be modified by the get and set attribute functions of the class. | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0.5 of 0.5** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 8. | Question : | In each of the following loops, determine the value of ires at the end of the loops execution and the number of times the loop executed:       a.     int ires = 1;             for (index = -2; index <= 2; index ++)                  ires += 2;        b.   int  ires = 1;           while (ires % 3 != 0) {                 ires += 3;           } | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | a. index is not initialized and will not run: index cannot be resolved to a variable     after initializing index by placing int in front of index and additionally by adding the brackets in order for the loop to work ie:     int ires = 1;             for (int index = -2; index <= 2; index ++){                  ires += 2;}    value = 11    count = 5  b. value = -2147483646     count = 715827883 | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 9. | Question : | Provide the output of the following Java program, given the input of: 5, 12, 6, and 23.  import java.text.\*;  import java.io.\*;   public class JavaInput {    public static void main (String argv []) throws IOException {       double result;       int int1, int2, int3, int4;        BufferedReader stdin =                 new BufferedReader (new InputStreamReader (System.in));       String UserInput = stdin.readLine ();       String [] list = UserInput.split (“\\+”);       int1 = Integer.parseInt (list [0]);       int2 = Integer.parseInt (list [1]);       int3 = Integer.parseInt (list [2]);       int4 = Integer.parseInt (list [3]);        result = (int1 + int2 + int3 + int4) / 4;       System.out.println ("The average is: " + result);     }  } | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | By utilizing the character "+" to separate the integers the output would be: The average is: 11.0 | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 10. | Question : | Provide the output for the following program:  import java.io.\*; public class Rocky {     public static void main (String [] args) {         String phrase = "Rocky, Bullwinkle, Friends and Everybody";         String mutation1, mutation2, mutation3, mutation4;         System.out.println ("Original string: \"" + phrase + "\"");         System.out.println ("Length of string: " + phrase.length ());          mutation1 = phrase.substring (0,5);         System.out.println ("Mutation #1: " + mutation1);         mutation2 = phrase.substring (7,25);         System.out.println ("Mutation #2: " + mutation2);         mutation1 = mutation1.concat (": The flying squirrel");         System.out.println ("Mutation #1b: " + mutation1);         mutation3 = mutation2.concat (" from Frostbite Falls");         System.out.println ("Mutation #3: " + mutation3);     } } | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Original string: "Rocky, Bullwinkle, Friends and Everybody" Length of string: 40 Mutation #1: Rocky Mutation #2: Bullwinkle, Friend Mutation #1b: Rocky: The flying squirrel Mutation #3: Bullwinkle, Friend from Frostbite Falls | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **4 of 4** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 11. | Question : | Write a complete *Java program* that will read as input the radius of a sphere (floating point) then calculate ***and***output the volume and Surface area Use the value Math.PI as the value for pi. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | import java.math.\*; import java.util.Scanner; public class Sphere { static Scanner mynextscanner = new Scanner(System.in); static float radius = 0; static float surfaceArea = 0; static float volume = 0; public static void main (String[]args){ getRadius(); getSurfaceArea(); getVolume(); System.out.println("The Sphere's radius is: " + radius); System.out.println("The Volume is: " + volume); System.out.println("The Surface Area is: " + surfaceArea); } public static void getRadius(){ System.out.println("Please input the radius for the sphere:"); radius = mynextscanner.nextFloat(); } public static void getSurfaceArea(){ surfaceArea = (float)(4\*Math.PI\*(Math.pow(radius, 2)));  } public static void getVolume(){ volume = (float)((4/3)\* Math.PI \* Math.pow(radius, 3 ));  }   } | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **4 of 4** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 12. | Question : | Write a java program to accept two integer values representing a range as parameters. Issue an error message if the second value is less than the first and terminate. Otherwise calculate and return the sum of the integer values in that range (inclusive). ie: if the input values are 5 and 10 the sum of the integer values is 45 (5 + 6 + 7 + 8 + 9 + 10). | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | import java.util.Scanner; public class sum {     public static void main(String [] args){     Scanner myscanner = new Scanner(System.in); int startrange = 0; int endrange = 0; System.out.println("Please input the beginning of the range."); startrange = myscanner.nextInt(); System.out.println("Please input the end of the range."); endrange = myscanner.nextInt(); int sum = 0; for (int i = startrange;i<endrange+1;i++){ sum = sum+i; } System.out.println("The sum is = " + sum); myscanner.close();     }  } | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3 of 4** | |  | Comments: | issue an error message if the second value is less than the first and terminate. - missing | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 13. | Question : | Write a complete Java program to read in a line of text, print out the individual words, each on a separate line. then print the total number of characters read and the number of words read (Note: look at the String and array methods to help you produce the output). | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | import java.util.ArrayList; import java.util.List; import java.util.Scanner; public class Stringinput {     public static void main(String [] args){     Scanner myscanner = new Scanner(System.in); String input; int characters; System.out.println("Please input your String."); input = myscanner.nextLine(); characters = input.length(); stringList parts = new stringList(); parts.insert(input.split("// "));    System.out.println("The String,"+ input); System.out.println("Is broken into "); parts.printList(); System.out.println("And contains " + characters + " Characters."); myscanner.close();     } public class stringList { private String List [];   private int currentCount;   private int maxSize;       public stringList(){   /\*\* initializes the list to contain a maximum of 10 items \*\*/   this.List = new String[10];   this.currentCount = 0;   this.maxSize = 10;   }   //Constructor with input   public stringList (int size){   /\*\* initializes the list to contain a maximum of size items \*\*/   this.List = new String[size];   this.currentCount = 0;   this.maxSize = size;   }   //Insert the string   public void insert (String[] strings){   /\*\* insert newString into the list into the next available position, if the list is full do nothing and discard newString \*\*/   boolean testvar = this.isFull();   if (testvar == false){   this.List[currentCount] = strings;   currentCount++;   }   else {   System.out.println("I'm Sorry I cannot Insert your value into the Array because it is full");   }   }   //Determine if the list is full   public boolean isFull(){   /\*\* return true if the list is full \*\*/   if(this.currentCount==this.maxSize && !this.List[currentCount].isEmpty() ){   return true;   }   else{   return false;   }   }   //Determine if the list is empty   public boolean isEmpty(){   /\*\* return true if the list is empty \*\*/   if (this.currentCount==0 && this.List[currentCount].isEmpty()){   return true;   }   else {   return false;   }   }   //Sort the list using an insertion sort   public void sortList (){   /\*\* sort the contents of the list \*\*/   if (this.isEmpty()){   System.out.println("I'm sorry, but I really don't feel comfortable sorting an empty list.....its' counterproductive");   }   else {   if (this.isFull()){   for (int i = 1; i<this.maxSize; i++){   String temp = this.List[i];   int j = i -1;   while (j > 0 && this.List[j].compareTo(temp)> 0){   this.List[j+1] = this.List[j];   j--;   }   this.List[j+1]= temp;   }   }   else {   for (int i= 1; i<this.currentCount; i++ ){   String temp = this.List[i];   int j = i -1;   while (j > 0 && this.List[j].compareTo(temp)> 0){   this.List[j+1] = this.List[j];   j--;   }   this.List[j+1]= temp;   }   }   }   }   //Print the list   public void printList (){   /\*\* print the entire contents of the list \*\*/   System.out.println("The Array has : " + this.maxSize + " Elements in the array.");   System.out.println("The elements of the Array are as follows: ");   for (int i= 0; i < this.maxSize; i++){   System.out.println("Element with index: " + i + " contains: " + this.List[i] );   }   }   //Clear the List   public void clearList(){   this.List = new String[this.maxSize];   }   //Set the current count at the end   public void setCurrentCountAtEnd(){   this.currentCount = this.currentCount - 1;   }  } | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **3.98 of 4** | |  | Comments: | The insert does not really work - the program has a list of strings in which the insert is adding on one position another array of strings (either add each element one by one or use a matrix private String List [][]) | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 14. | Question : | Implement (define) a class named StockType.java. A StockType has a symbol (a String), a type (an int) and a price (a double). It provides the methods to set the price (give the price an initial value), change the price (increase or decrease the price by some value), to return the current price, to set the symbol value, to return the symbol value, to set the type value and return the type value. Objects that are declared to be of StockType are given an initial price of 0, an initial type of 0 and an initial symbol value of the empty string. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | public class StockType { private String symbol; private int type; private double price; public StockType(){ this.symbol = ""; this.type = 0; this.price = 0; } void setSymbol(String input){ this.symbol = input; } void setType(int input){ this.type = input; } void addToPrice(double input){ this.price = this.price + input; } void subtractFromPrice(double input){ this.price = this.price - input; } String getSymbol(){ return this.symbol; } int getType(){ return this.type; } double getPrice(){ return this.price; }     public static void main (String [] args) {          } } | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **4.5 of 4.5** | |  | Comments: |  | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Question 15. | Question : | Given a Java int array containing the following values:   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 3 | 37 | 13 | 25 | 21 | -3 | 22 | 11 | 41 | -5 |   What is the order of the values following the first pass of the selection sort we used in class:   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |   What is the order of the values following the second pass of the selection sort:   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |   What is the order of the values following the third pass of the selection sort:   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  |   What is the order of the values following the fourth pass of the selection sort:   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  |  | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | Student Answer: |  | Original   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 3 | 37 | 13 | 25 | 21 | -3 | 22 | 11 | 41 | -5 |   First Pass   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | -5 | 37 | 13 | 25 | 21 | -3 | 22 | 11 | 41 | 3 |   Second Pass   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | -5 | -3 | 13 | 25 | 21 | 37 | 22 | 11 | 41 | 3 |   Third Pass   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | -5 | -3 | 3 | 25 | 21 | 37 | 22 | 11 | 41 | 13 |   Fourth Pass   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | -5 | -3 | 3 | 11 | 21 | 37 | 22 | 25 | 41 | 13 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **4 of 4** | |  | Comments: |  | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Question 16. | Question : | Given a Java String array containing the following values:   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Alan | Bernard | Casper | Daniel | Dennis | Herb | Micheal | Stephen | Wilber | Zelda |   a) How many probes of the binary search are required to locate Alan? (Explain how did you reach to that result).  b) How many probes of the binary search are required to determine Mary is not in the list? (Explain how did you reach to that result). | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | With my definition being that a probe is every evaluation of the variable: a.) Arraysize /2 = Dennis      is Dennis = Alan:: No, count 1      is Dennis > Alan:: Yes, Eliminate all cells greater than Dennis count 2      Arraysize /2 = Casper     is Casper = Alan:: No, count 3     is Casper > Alan:: Yes, Eliminate all cells greater than Casper count 4     Arraysize /2 = Bernard     is Bernard = Alan:: No, count 5     is Bernard > Alan:: Yes, Eliminate all cells greater than Bernard     Arraysize <= 2      Remaining cell = Alan     is Alan = Alan::Yes, Return Index 0 Count 6     Count = 6  b.) Arraysize /2 = Dennis      is Dennis = Mary:: No, count 1      is Dennis > Mary:: No, Eliminate all cells less than Dennis count 2      Arraysize /2 = Michael     is Michael = Mary:: No, count 3     is Michael > Mary:: Yes, Eliminate all cells greater than Michael count 4     Arraysize <= 2      Remaining cell = Herb     is Herb = Mary:: No, Return Index -1 Count 5     Count = 5 | |  | Instructor Explanation: | Each question is 2 points | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **4 of 4** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 17. | Question : | What is the characteristic behavior of a stack? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | A characteristic behavior of a stack is that information can be input on a First in Last out basis. Meaning that information is only available that has just been pushed into the stack.  ie. the input 1,2,3,4,5 would be available as 5,4,3,2,1 and that not variables can be independently associated. | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **1 of 1** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 18. | Question : | What are the five basic operations on a stack? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Push, pop, peek, empty and search | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **1 of 1** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 19. | Question : | What is difference between a stack and a queue? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | The main difference between a stack and a queue is that information input into a queue is available for output in a different way. Queues operate on a First in First out methodology meaning that the information that was input is always referenced and will always be utilized first. ie. the input of 1,2,3,4,5 will be available in the order 1,2,3,4,5 | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **1 of 1** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | Question 20. | Question : | What are the five basic operations on a queue? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Add, offer, remove, poll, element, peek | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **1 of 1** | |  | Comments: |  | | | |