|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

The result in a Java program of the expression 12/5 is integer 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | True | | |
|  |  | False | | |
| **Question 2** | | |  | 1 / 1 point | |

What will be in the variable num after the following Java statement executes?

double num = 12/5;

|  |  |  |  |
| --- | --- | --- | --- |
|  | Nothing - it won't compile | | |
|  | 2.4 | | |
|  | 2.0 | | |
|  | None of the other answers | | |
| **Question 3** | |  | 1 / 1 point | |

What will be in the variable num after the following Java statement executes?

double num = 12/5.0;

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | Nothing - the statement will not compile |
|  | | | 2.4 |
|  | | | 2.0 |
|  | | | None of the other answers |
|  |  |
|  | |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 0 / 1 point |

Given

public class Numbers {  
 private int [] numbers = new int[5];  
  
    public Numbers() {  
    }  
  
    public Numbers(size) {  
    numbers = new int[size];  
    }

and in ***main***, we have declared

Numbers num = new Numbers(22);

How many ***int***locations are in memory associated with the object ***num***?

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | | |
|  | 5 | | |
|  | 10 | | |
|  | 22 | | |
| **Question 2** | |  | 1 / 1 point | |

What displays from the following statement?

int num = 4;  
  
if (num >= 5)   
    System.out.println ("statement 1");  
else  
    System.out.println ("statement 2");

|  |  |  |  |
| --- | --- | --- | --- |
|  | statement 1 | | |
|  | statement 2 | | |
|  | nothing | | |
|  | none of the above | | |
| **Question 3** | |  | 1 / 1 point | |

What displays from the following statement?

int num = 4;  
while (num >= 5)  {  
    System.out.println ("statement 1");  
}  
System.out.println ("statement 2");

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | statement 1 |
|  | | | statement 2 |
|  | | | statement 1 repeated forever |
|  | | | statement 1  statement 2 |
|  |  |
|  | |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Question 1** |  | 1 / 1 point |

What would display from:

double x = 5.0 + 3/2;

System.out.println (x);

|  |  |  |  |
| --- | --- | --- | --- |
|  | 4.0 | | |
|  | 6.5 | | |
|  | 6.0 | | |
|  | won't compile | | |
| **Question 2** | |  | 0 / 1 point | |

Given the declaration  Integer [] nums = new Integer[5];

If a reference uses 1 byte in memory, and an Integer object uses 4 bytes in memory, how much memory is allocated with the above declaration?

|  |  |  |  |
| --- | --- | --- | --- |
|  | 6 | | |
|  | 20 | | |
|  | 21 | | |
|  | 26 | | |
| **Question 1** | |  | 1 / 1 point |

What is the result of the following expression in Java

10%3 + 2

|  |  |  |  |
| --- | --- | --- | --- |
|  | 3 | | |
|  | 5.33333 | | |
|  | 5 | | |
|  | none of above | | |
| **Question 2** | |  | 1 / 1 point | |

What is the result of the following expression in Java

12/5.0 + 6/4

|  |  |  |  |
| --- | --- | --- | --- |
|  | 3.9 | | |
|  | 3.4 | | |
|  | 3 | | |
|  | none of above | | |
| **Question 3** | |  | 0 / 1 point | |

Given    ***public class Numbers {***

***private int [] numbers = new int[5];***

***public Numbers() {***

***}***

***public Numbers(size) {***

***numbers = new int[size];***

***}***

***public boolean doSomething() {***

***if (numbers.length >= 2)***

***return (numbers[0] > numbers[1])***

***else return false;***

***}***

Will the method ***doSomething()*** compile?

|  |  |  |  |
| --- | --- | --- | --- |
|  | yes - and it will compare the values in the two elements | | |
|  | no - there's a missing if statement around the comparison of ***numbers[0]***and ***numbers[1]*** | | |
|  | no - you can't compare two elements of the array  using > | | |
|  | yes - but it will not compare the values in the two elements, it will compare the reference values - ie where they are in memory | | |
|  | none of above | | |
| **Question 4** | |  | 1 / 1 point | |

Given    ***public class Dates {***

***private MyDate [] dates = new MyDate[5];***

***public Dates() {***

***}***

***public Dates(size) {***

***dates= new MyDate[size];***

***}***

***public boolean doSomething() {***

***if (dates.length >= 2)***

***return (dates[0] > dates[1])***

***else return false;***

***}***

Will the method ***doSomething()*** compile?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | yes - and it will compare the values in the two elements | | |
|  | | | no - there's a missing if statement around the comparison of ***dates[0]***and ***dates[1]*** | | |
|  | | | no - you can't compare two elements of the array  using > | | |
|  | | | yes - but it will not compare the values in the two elements, it will compare the reference values - ie where they are in memory | | |
|  | | | none of the above | | |
|  |  |
| **Question 1** | | | |  | 1 / 1 point |

What is the value of num after this code executes?

int num = 1;  
for (int i = 0; i < 3; i++)   
      for (int j= 0; j < i; j++)  
          num+= i+j;

|  |  |  |  |
| --- | --- | --- | --- |
|  | 15 | | |
|  | 8 | | |
|  | 6 | | |
|  | 7 | | |
| **Question 2** | |  | 1 / 1 point | |

Out of the following choices, which is the BEST  bigO() for an algorithm?

|  |  |  |  |
| --- | --- | --- | --- |
|  | O(n2) | | |
|  | O(n) | | |
|  | O(nlogn) | | |
|  | O(log n) | | |
| **Question 3** | |  | 1 / 1 point | |

Out of the following choices, which is the WORST  bigO() for an algorithm?

|  |  |  |  |
| --- | --- | --- | --- |
|  | O(n2) | | |
|  | O(n) | | |
|  | O(nlogn) | | |
|  | O(logn) | | |
| **Question 4** | |  | 0 / 1 point | |

What is the bigO of the following pseudocode?

for (i=0; i<n; i++)  
      for (j=n; j>0; j/=2)  
           print something  
      end for  
end for  
  
for (i=0; i<n; i+=2)  
      print something  
end for

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | O(logn) |
|  | | | O(nlogn) |
|  | | | O(n2 log n) |
|  | | | none of these answers |
|  |  |
|  | |

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 1 / 1 point |

Given the declaration:    **int [] numbers= new int[10];**

**numbers** is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | an array of  10 ints | | | |
|  | |  |  | | --- | --- | |  | a reference to an array of 10 ints | | | |
|  | |  |  | | --- | --- | |  | a reference to an array of 9 ints | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 2** | |  | 1 / 1 point | |

Given the declaration int [ ] nums = {8, 12, 23, 4, 15}, what expression will display the first element in the array (ie the number 8)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | System.out.print("The number is : " + nums[0]); | | | |
|  | |  |  | | --- | --- | |  | System.out.print("The number is : " + nums[1]); | | | |
|  | |  |  | | --- | --- | |  | System.out.print("The number is : " + nums[8]); | | | |
| **Question 3** | |  | 1 / 1 point | |

Given the declaration : int [ ] ar = {1,2,3,4,5}; What is the value of ar[3]?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 2 | | | |
|  | |  |  | | --- | --- | |  | 3 | | | |
|  | |  |  | | --- | --- | |  | 4 | | | |
|  | |  |  | | --- | --- | |  | 5 | | | |
| **Question 4** | |  | 1 / 1 point | |

If we declare int [ ] ar = {1,2,3,4,5,6}; The size of array ar is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 0 | | | |
|  | |  |  | | --- | --- | |  | 5 | | | |
|  | |  |  | | --- | --- | |  | 6 | | | |
|  | |  |  | | --- | --- | |  | 7 | | | |
| **Question 5** | |  | 1 / 1 point | |

The last value in an array called ar can be found at index:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 0 | | | |
|  | |  |  | | --- | --- | |  | 1 | | | |
|  | |  |  | | --- | --- | |  | ar.length | | | |
|  | |  |  | | --- | --- | |  | ar.length - 1 | | | |
| **Question 6** | |  | 1 / 1 point | |

The most common use of an array is to:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | perform for loop on array | | | |
|  | |  |  | | --- | --- | |  | perform different operations on each element in array | | | |
|  | |  |  | | --- | --- | |  | perform the same operation on all elements in array | | | |
| **Question 7** | |  | 1 / 1 point | |

The range of indices for an array always start at:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | whatever programmer specifies | | | |
|  | |  |  | | --- | --- | |  | 1 | | | |
|  | |  |  | | --- | --- | |  | 0 | | | |
|  | |  |  | | --- | --- | |  | size of array | | | |
| **Question 8** | |  | 1 / 1 point | |

What loop will display each of the numbers in this array on a separate line: float [ ] nums= {1.1f, 2.2f, 3.3f};

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | for (int i =0; i < 3; i++) System.out.println( nums[i]); | | | |
|  | |  |  | | --- | --- | |  | for (i = 1; i <= 3; i++) System.out.println(nums[i]); | | | |
|  | |  |  | | --- | --- | |  | for (i = 0; i <= 3; i++) System.out.println(nums[i]); | | | |
|  | |  |  | | --- | --- | |  | for (i = 1; i < 3; i++) System.out.println(nums[i]); | | | |
| **Question 9** | |  | 1 / 1 point | |

What would display from the following statements? int [ ] nums = {1,2,3,4,5,6}; System.out.println((nums[1] + nums[3]));

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 1 + 3 | | | |
|  | |  |  | | --- | --- | |  | 4 | | | |
|  | |  |  | | --- | --- | |  | 2 + 4 | | | |
|  | |  |  | | --- | --- | |  | 6 | | | |
| **Question 10** | |  | 1 / 1 point | |

Assume we have written the Date class, what statement will declare an object of Date class and initialize the date in that object to Nov 25, 2008.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | date dateObj(11, 25, 2008); | | | |
|  | |  |  | | --- | --- | |  | Date dateObj(11, 25, 2008); | | | |
|  | |  |  | | --- | --- | |  | Date dateObj = {11, 25, 2008}; | | | |
|  | |  |  | | --- | --- | |  | Date dateObj = new Date(11, 25, 2008); | | | |
| **Question 11** | |  | 1 / 1 point | |

A default constructor has how many parameters?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 0. | | | |
|  | |  |  | | --- | --- | |  | 1. | | | |
|  | |  |  | | --- | --- | |  | 2. | | | |
|  | |  |  | | --- | --- | |  | Variable. | | | |
| **Question 12** | |  | 1 / 1 point | |

A subclass (or derived class)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | incorporates data fields and methods from its superclass and can have additional fields and methods as well | | | |
|  | |  |  | | --- | --- | |  | incorporates data fields and methods from its superclass but cannot have additional fields and methods | | | |
|  | |  |  | | --- | --- | |  | incorporates data fields , but not methods, from its superclass and can have additional fields or methods as well | | | |
|  | |  |  | | --- | --- | |  | incorporates data fields, but not methods, from its superclass but cannot have additional fields or methods | | | |
| **Question 13** | |  | 1 / 1 point | |

An algorithm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | is similar to logarithm | | | |
|  | |  |  | | --- | --- | |  | concentrates on the syntactic solution to a problem | | | |
|  | |  |  | | --- | --- | |  | is a step-by-step solution to a problem written in English | | | |
|  | |  |  | | --- | --- | |  | is not needed | | | |
| **Question 14** | |  | 0 / 1 point | |

At Algonquin College we will use PDL to express an

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Answer: | Program Development Language | Incorrect Response**(algorithm, problem solution, solution, program)** | | |
| **Question 15** | | |  | 1 / 1 point | |

Does declaring a catch of a superclass exception catch all of the subclass exceptions of that superclass?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1) True | | |
|  |  | 2) False | | |
| **Question 16** | | |  | 1 / 1 point | |

Every Java application is composed of at least one:

|  |  |  |  |
| --- | --- | --- | --- |
|  | local variable | | |
|  | public interface declaration | | |
|  | public class declaration | | |
|  | imported class | | |
| **Question 17** | |  | 1 / 1 point | |

Given String sName = "Joe Blow"; What displays from System.out.println (sName.charAt(2));

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | o | | | |
|  | |  |  | | --- | --- | |  | e | | | |
|  | |  |  | | --- | --- | |  | statement will not compile | | | |
|  | |  |  | | --- | --- | |  | statement will cause a run-time error | | | |
| **Question 18** | |  | 1 / 1 point | |

Given String sName = "Joe Blow"; What is the value returned by sName.length()?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | 7 | | | |
|  | |  |  | | --- | --- | |  | 8 | | | |
|  | |  |  | | --- | --- | |  | 9 | | | |
|  | |  |  | | --- | --- | |  | 10 | | | |
| **Question 19** | |  | 1 / 1 point | |

Given String sName = "Joe Blow"; What will result if we execute the expression if (sName.compareTo("Joe Blow") == 0)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | result will be true because both sides of the == contain the value "Joe Blow", and compareTo method will return a 0 in that case. | | | |
|  | |  |  | | --- | --- | |  | result will be false because the reference value stored in sName will not be the same as the reference value for the literal "Joe Blow" | | | |
|  | |  |  | | --- | --- | |  | expression will not compile | | | |
|  | |  |  | | --- | --- | |  | expression will cause a run-time error | | | |
| **Question 20** | |  | 1 / 1 point | |

Given String sName = "Joe Blow"; Will this compile?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Yes | | |
|  | |  |  | | --- | --- | |  | No - but String sName = new String ("Joe Blow"); will fix it. | | |
|  | |  |  | | --- | --- | |  | No - but String sName = new String(); followed by sName = in.nextLine() (where in is an object of type Scanner) will fix it. | | |
|  | |  |  | | --- | --- | |  | No | | |
| [View Feedback](javascript://) | |

|  |  |  |
| --- | --- | --- |
| **Question 21** |  | 1 / 1 point |

Given String sName = "Joe Blow"; can we execute sName = "Sally Sue Smith";?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Yes - the reference value for the literal "Sally Sue Smith" is copied into sName (which is a reference to a String). | | |
|  | |  |  | | --- | --- | |  | No - the size of the String is allocated when it is created and "Sally Sue Smith" is too big. | | |
|  | |  |  | | --- | --- | |  | No - Strings are immutable, hence cannot be changed. | | |
|  | |  |  | | --- | --- | |  | none of above | | |
| [View Feedback](javascript://) | |

|  |  |  |
| --- | --- | --- |
| **Question 22** |  | 1 / 1 point |

Given String sName; Which statement will initialize sName to "Joe Blow"?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | sName = in.nextLine(); where in is an object of type Scanner, and the user types in Joe Blow | | |
|  | |  |  | | --- | --- | |  | sName = new String ("Joe Blow"); | | |
|  | |  |  | | --- | --- | |  | sName = "Joe Blow"; | | |
|  | |  |  | | --- | --- | |  | All of the above | | |
| [View Feedback](javascript://) | |

|  |  |  |
| --- | --- | --- |
| **Question 23** |  | 0 / 1 point |

Given the declaration of an object of class Person called person1; and that Person class has a field called nAge, would we expect the following statement to be allowed: System.out.println( "Age is " + person1.nAge );

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Incorrect Response | |  |  | | --- | --- | |  | yes | | | |
| Correct Answer | |  |  | | --- | --- | |  | no | | | |
| **Question 24** | |  | 1 / 1 point | |

Given the declaration: Person person1;....A call to displayPerson method would be:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | Person.displayPerson(); | | | |
|  | |  |  | | --- | --- | |  | person1.displayPerson; | | | |
|  | |  |  | | --- | --- | |  | Person.displayPerson; | | | |
|  | |  |  | | --- | --- | |  | person1.displayPerson(); | | | |
| **Question 25** | |  | 0 / 1 point | |

Inheritance refers to the ability to

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Incorrect Response | |  |  | | --- | --- | |  | redefine how member methods of related classes operate | | | |
| Correct Answer | |  |  | | --- | --- | |  | create new classes from existing ones | | | |
|  | |  |  | | --- | --- | |  | combine data fields and methods into one type | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 26** | |  | 0 / 2 points | |

Select all the conditions that should be considered in a test plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Correct Answer | |  |  | | --- | --- | |  | valid values for each input in the problem solution | | | |
| Correct Answer | |  |  | | --- | --- | |  | invalid values for each input in the problem solution | | | |
| Correct Answer | |  |  | | --- | --- | |  | result of calculations of all valid input values | | | |
| Correct AnswerIncorrect Response | |  |  | | --- | --- | |  | boundary conditions of values for each input (one less than boundary, the actual boundary and one more than the boundary) | | | |
| Correct Answer | |  |  | | --- | --- | |  | Potential conditions that could cause the solution to have problems (for example division by 0) | | | |
| **Question 27** | |  | 0 / 1 point | |

Select keywords that are used in PDL

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | if, then, else, while, for, dowhile, repeat, put, get | | | |
|  | |  |  | | --- | --- | |  | IF, THEN, ELSE, WHILE, FOR, DOWHILE, REPEAT, PUT, GET | | | |
| Correct Answer | |  |  | | --- | --- | |  | PUT or DISPLAY, GET or INPUT, IF, ELSE, ENDIF, WHILE, ENDWHILE | | | |
| Incorrect Response | |  |  | | --- | --- | |  | none of the above. | | | |
| **Question 28** | |  | 0.5 / 3 points | |

Put the following steps to solving a problem in order:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Incorrect Response | \_\_5\_\_ | **(1)** | Run and debug the program | | |
| Incorrect Response | \_\_6\_\_ | **(1)** | Install and maintain the program | | |
| Incorrect Response | \_\_2\_\_ | **(1)** | Outline solution into an algorithm with PDL | | |
| Incorrect Response | \_\_4\_\_ | **(1)** | Test algorithm solution for correctness | | |
| Incorrect Response | \_\_3\_\_ | **(1)** | Code algorithm into a program | | |
|  | \_\_1\_\_ |  | Define the problem | | |
| **Question 29** | | | |  | 1 / 1 point |

The difference between a checked exception and unchecked exception is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | The compiler verifies that all checked exceptions thrown by each method are caught in the calling code (or declared in a throw clause). | | | |
|  | |  |  | | --- | --- | |  | The JVM verifies that all checked exceptions thrown by each method are caught in the calling code (or declared in a throw clause) | | | |
|  | |  |  | | --- | --- | |  | Java catches all checked exceptions automatically if the programmer has not done so. | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 30** | |  | 1 / 1 point | |

The exception thrown by the Scanner class when executing method nextInt, and invalid chars are entered for the int is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | ArithmeticException | | | |
|  | |  |  | | --- | --- | |  | IndexOutOfBoundsException | | | |
|  | |  |  | | --- | --- | |  | InputMismatchException | | | |
|  | |  |  | | --- | --- | |  | IOException | | | |
| **Question 31** | |  | 1 / 1 point | |

The exception thrown when the program executes a division where the denominator is 0 is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | IOException | | | |
|  | |  |  | | --- | --- | |  | ArithmeticException | | | |
|  | |  |  | | --- | --- | |  | InputMismatchException | | | |
|  | |  |  | | --- | --- | |  | none of the above | | | |
| **Question 32** | |  | 1 / 1 point | |

The following prototype shows that a Cylinder subclass is derived from a superclass called Circle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | class Circle extends Cylinder | | | |
|  | |  |  | | --- | --- | |  | class Cylinder derived Circle | | | |
|  | |  |  | | --- | --- | |  | class Cylinder extends Circle | | | |
|  | |  |  | | --- | --- | |  | class Circle derived Cylinder | | | |
| **Question 33** | |  | 2 / 2 points | |

The purpose of a structured approach to solving a problem is: (select all that apply)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | to decrease development time as you will have fewer programming and logic errors | | | |
|  | |  |  | | --- | --- | |  | to have a strategy that can be used to solve complex problems | | | |
|  | |  |  | | --- | --- | |  | to make program maintenance easier due to better written programs | | | |
|  | |  |  | | --- | --- | |  | to annoy students who think (incorrectly) that this takes more time than its worth. | | | |
| **Question 34** | |  | 1 / 1 point | |

The purpose of exception handling (using try...catch) is to handle common errors in your program.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1) True | | |
|  |  | 2) False | | |
| **Question 35** | | |  | 1 / 1 point | |

True or false. Reference-type variables are initialized by default to the value null.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | |  | true | | | |
|  | |  |  | | --- | --- | |  | false | | | |
| **Question 36** | |  | 39 / 39 points | |

Method overloading and method overriding are the two kinds of polymorhism. You get a bonus if you select 'True' because its true.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | True |
|  |  | | False |
|  |  |
|  | |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Question 1** |  | 0 / 5 points |

Without writing this code into Eclipse and running it, what would be output from the following:

public class TestStuff {  
 public static void main(String[] args) {  
 Stuff myStuff = new Stuff(12);  
 System.out.println(myStuff);  
 myStuff.doSomething(6.4);  
 System.out.println(myStuff);  
 String s = "Original";  
 myStuff.ChangeString(s);  
 System.out.println(s);  
 System.out.println(myStuff);  
 }    
}  
  
public class Stuff {  
 private static final double *x* = 0.8;  
 private int n;  
 private String s = "Not Set";  
  
 public Stuff(int n) {        
 this.n = n / 5;     
 }  
  
 public void ChangeString(String s) {  
        s = "Revised";  
        System.out.println(s);  
    }  
  
    public void doSomething(double d) {  
     n = (int) d \* 2;  
        System.out.println(this);  
        n = (int) doSomeMore();  
    }  
  
    private double doSomeMore() {  
        double d = n \* *x*;  
        System.out.println(d);  
        return d;  
    }  
  
    public String toString() {  
        return "Stuff has " + n + " and " + s;  
    }  
}

Stuff has 2 and Not Set  
Stuff has 12 and Not Set  
9.60000000000000  
Stuff has 9 and Not Set  
Revised  
Original  
Stuff has 9 and Not Set

**This question has not been graded.**

**The correct answer is not displayed for Written Response type questions.**

|  |  |  |
| --- | --- | --- |
| **Question 2** |  | 5 / 5 points |

Overloading and Overriding are two ways of implementing Polymorphism in Java.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | True |
|  |  | | False |
|  |  |
|  | |

.

|  |  |  |
| --- | --- | --- |
| **uestion 1** |  | 1 / 1 point |

What would display from:

double x = 5.0 + 3/2;

System.out.println (x);

|  |  |  |  |
| --- | --- | --- | --- |
|  | 4.0 | | |
|  | 6.5 | | |
|  | 6.0 | | |
|  | won't compile | | |
| **Question 2** | |  | 0 / 1 point | |

Given the declaration  Integer [] nums = new Integer[5];

If a reference uses 1 byte in memory, and an Integer object uses 4 bytes in memory, how much memory is allocated with the above declaration?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | 6 | | |
|  | | | 20 | | |
|  | | | 21 | | |
|  | | | 26 | | |
|  |  |
|  | |
| **Question 1** | | | |  | 0 / 1 point |

What displays from this code segment when n=6?

for (int i = 0; i < n; i++)   
 System.out.print (i + “ ”);  
System.out.println ();  
for (int j=n; j > 0; j--)  
 System.out.print ( j + “ “);

0 1 2 3 4 5  
6 5 4 3 2 1

**This question has not been graded.**

**The correct answer is not displayed for Written Response type questions.**

|  |  |  |
| --- | --- | --- |
| **Question 2** |  | 0 / 1 point |

What displays from this code segment when n=6?

for (int i = 0; i < n; i++) {   
 System.out.print (i + “ ”);  
 for (int j=n; j > i ; j--)  
 System.out.print ( j + “ “);  
 System.out.println();  
}

0 6 5 4 3 2 1  
1 6 5 4 3 2  
2 6 5 4 3  
3 6 5 4  
4 6 5  
5 6

**This question has not been graded.**

**The correct answer is not displayed for Written Response type questions.**

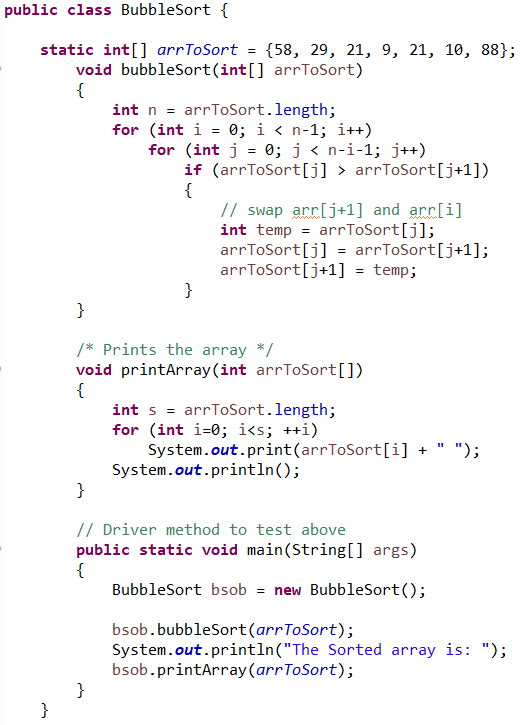
|  |  |  |
| --- | --- | --- |
| **Question 3** |  | 0 / 1 point |

What displays from this code segment when n=17?

for (int i=1; i < n; i\*=2) {  
 int j = n;   
 while ( j>0) {  
 System.out.println (“ “ + i + “ “ + j );  
 j-=2;  
 }  
 System.out.print (“\n”);  
}

1 17  
1 15  
1 13  
1 11  
1 9  
1 7  
1 5  
1 3  
1 1  
  
2 17  
2 15  
2 13  
2 11  
2 9  
2 7  
2 5  
2 3  
2 1  
  
4 17  
4 15  
4 13  
4 11  
4 9  
4 7  
4 5  
4 3  
4 1  
  
8 17  
8 15  
8 13  
8 11  
8 9  
8 7  
8 5  
8 3  
8 1  
  
16 17  
16 15  
16 13  
16 11  
16 9  
16 7  
16 5  
16 3  
16 1

|  |  |  |
| --- | --- | --- |
| **Question 1** |  | 0 / 5 points |



|  |  |  |
| --- | --- | --- |
| Answer: | 9 10 21 21 29 58 88 | Incorrect Response**(The Sorted array is: 9 10 21 21 29 58 88)** |
| Abstraction  Representation of something that leaves out a lot of detail to better emphasize some aspect for study or better use.  O(1)  Consider a queue implemented by array and containing N elements. The deque operation has the worst case time complexity of what?  O(1)  Consider a stack implemented with linked cells and containing N elements. The push operation has worst case time complexity of what?  O(1)  Consider a stack implemented with an array and containing N elements. The push operation has worst case time complexity of what?  Anytime In Never Out (AINO)  The behavior of the following axiom is best described as what?  remove(add(DS, elt)) = add(DS, elt)  First In First Out (FIFO)  The behavior described by the following axiom is best described as what?  remove(add(DS, elt)) = ite(DS = new, new, add(remove(DS), elt)  Last In First Out (LIFO)  The behavior of the following axiom is best described as what?  remove(add(DS, elt)) = DS  O(N)  Consider a list implemented by an array and containing N elements. The add operation has best case time complexity of what?  O(N)  Consider a list implemented by linked cells, and containing N elements. The add operation has worst case time complexity of what?  O(N)  Consider a list implemented by array, and containing N elements. The remove operation has worst case time complexity of what?  O(1)  Consider a list implemented by array, and containing N elements. The get operation has worst case time complexity of what?  O(N)  Consider a list implemented by linked cells, and containing N elements. The get operation has worst case time complexity of what?  True  (T/F) O(10 \* N^3) is a valid description of an algorithm that is known to run in O(N^3).  False  (T/F) Is O(N^(N^1/2)) a valid description of an algorithm that is known to run in O(N^3)?  O((1/2) MAX^2)  Describe the time complexity of the following code. Your answer must be as precise as possible:  for (i=0; i<MAX; i++) {  for (k=0; k<=i; k++) {  arr[k,i] = 2;}}  True  (T/F) Is O(2^N) theoretically worse than O(N^2)?  False  (T/F) O((2N)^2) theoretically worse than O(N^2).  False  (T/F) O(2(N^2)) is theoretically worse than O(N^2).  False  (T/F) O((N+1)^2) is theoretically worse than O(N^2).  O(N^2)  Sorting a sequence of N numbers by adding them successively to a list (finding and putting them into the proper location by value) has worst case time complexity of what?  O(N^2)  What best describes the worst case time complexity of BubbleSort on a list of N elements ?  D, B, F, A, C, F, E, G  Given a perfectly inserted binary search tree containing elements A through G, what is the sequence from a breadth-first traversal?  D, B, A, C, F, E, G  Given a perfectly inserted binary search tree containing elements A through G, what is the sequence from a pre-order traversal?  A, B, C, D, E, F, G  Given a perfectly inserted binary search tree containing elements A through G, what is the sequence from a in-order traversal?  A, C, B, E, G, F, D  Given a perfectly inserted binary search tree containing elements A through G, what is the sequence from a post-order traversal?  2  Given a perfectly inserted binary search tree containing elements A through G, what is the height of the tree?  1  Given a perfectly inserted binary search tree containing elements A through G, what is the node depth of the beta?  1  Given a perfectly inserted binary search tree containing elements A through G, what is the node height of the beta?  In-Order Tranversal  To list the keys in a binary search tree in alphabetical order, the tree traversal to use is?  False  (T/F) For any particular set of keys (values), there is exactly one binary search tree that properly organizes those keys according to the BST ordering property.  16  A binary search tree has a height of 15. What is the least number of nodes that might be in that BST?  65535 (2^16 - 1)  A binary search tree has a height of 15. What is the greatest number of nodes that might be in that BST?  O(log N)  For a binary search tree containing N keys, inserting a new key has average case time complexity of what?  O(N)  For a binary search tree containing N keys, inserting a new key has worst case time complexity of what?  Garbage Collection  Dynamic memory allocated in the heap, but no longer reachable from a program, is returned to the unallocated heap.  Infinite  The amount of run-time stack space necessary to run the following code can be characterized as what?  function main() {  var x = 6;  var res = factorial(x)  print(res)  }  function factorial(n) {  if (n==1) return 1;  return n \* factorial(n-2);  }  Finite and Unbounded  The amount of run-time stack space necessary to run the following code can be characterized as what?  function main() {  var x = getUserInput();  var res = factorial(x)  print(res)  }  function factorial(n) {  if (n<=1) return 1;  return n \* factorial(n-1);  }  Infinite  The amount of run-time stack space necessary to run the following code can be characterized as what?  function main() {  var x = getUserInput();  var res = factorial(x)  print(res)  }  function factorial(n) {  if (n==1) return 1;  return n \* factorial(n-1);  }  Finite and Bounded  The amount of run-time stack space necessary to run the following code can be characterized as what?  function main() {  var x = 5;  var res = factorial(x)  print(res)  }  function factorial(n) {  if (n==1) return 1;  return n \* factorial(n-1);  }  O(1)  BigO array access average case?  O(N)  BigO array search average case?  O(N)  BigO array insert average case?  O(N)  BigO array delete average case?  O(1)  BigO array access worst case?  O(N)  BigO array search worst case?  O(N)  BigO array insert worst case?  O(N)  BigO array delete worst case?  O(N)  BigO stack access average case?  O(N)  BigO stack search average case?  O(1)  BigO stack insert average case?  O(1)  BigO stack delete average case?  O(N)  BigO stack access worst case?  O(N)  BigO stack search worst case?  O(1)  BigO stack insert worst case?  O(1)  BigO stack delete worst case?  O(N)  BigO queue access average case?  O(N)  BigO queue search average case?  O(1)  BigO queue insert average case?  O(1)  BigO queue delete average case?  O(N)  BigO queue access worst case?  O(N)  BigO queue search worst case?  O(1)  BigO queue insert worst case?  O(1)  BigO queue delete (dequeue) worst case? (Linked-list implementation)  O(N)  BigO queue delete (dequeue) worst case? (Array implementation)  O(N)  BigO list access average case?  O(N)  BigO list search average case?  O(1)  BigO list insert average case?  O(1)  BigO list delete average case?  O(N)  BigO list access worst case?  O(N)  BigO list search worst case?  O(1)  BigO list insert-cell worst case? (Linked-List implementation)  O(N)  BigO list insert worst case?  O(1)  BigO list delete-cell worst case? (Linked-List implementation)  O(N)  BigO list delete worst case?  O(log N)  BigO BST access average case?  O(log N)  BigO BST search average case?  O(log N)  BigO BST insert average case?  O(log N)  BigO BST delete average case?  O(N)  BigO BST access worst case?  O(N)  BigO BST search worst case?  O(N)  BigO BST insert worst case?  O(N)  BigO BST delete worst case?  O(log N)  BigO binary heap insert worst case?  O(1)  BigO binary heap getMin worst case?  O(log N)  BigO binary heap delMin worst case?  O(N)  BigO binary heap search worst case?  O(1) (~O(2.67))  BigO binary heap insert average case?  O(1)  BigO binary heap getMin average case?  O(log N)  BigO binary heap delMin average case?  O(N)  BigO binary heap search average case?  False  (T/F) A data abstraction includes details related to the implementation.  True  (T/F) A data abstraction includes changes made to state of the data when various operations are performed.  All of the operations, the types of the arguments of the operations, and the type of the return of the operation.  The functional signature of every ADT has these three characteristics.  Canonical Operation  This is needed if your goal is to generate ALL possible ADT values by calling successive operations.  new: -> QUEUE  push: STACK x Int -> STACK  pop: STACK -> STACK  top: STACK -> Int  size: STACK -> Nat  Signature of a stack.  new: -> QUEUE  enque: QUEUE x Int -> QUEUE  deque: QUEUE -> QUEUE  front: QUEUE -> Int  size: QUEUE -> Nat  Signature of a queue.  O(N)  BigO worst case insert operation for a list implemented by an array structure.  O(N)  BigO worst case remove operation for a list implemented by an array structure.  O(1)  BigO worst case get operation for a list implemented by an array structure.  O(N)  BigO worst case find operation for a list implemented by an array structure.  O(1)  BigO worst case empty operation for a list implemented by an array structure.  O(1)  BigO worst case size operation for a list implemented by an array structure.  O(N)  BigO worst case insert operation for a list implemented by a linked structure.  O(1)  BigO worst case insert-cell operation for a list implemented by a linked structure.  O(N)  BigO worst case remove operation for a list implemented by a linked structure.  O(1)  BigO worst case remove-cell operation for a list implemented by a linked structure.  O(N)  BigO worst case get operation for a list implemented by a linked structure.  O(N)  BigO worst case find operation for a list implemented by a linked structure.  O(1)  BigO worst case empty operation for a list implemented by a linked structure.  O(1)  BigO worst case size operation for a list implemented by a linked structure.  new: -> LIST  insert: LIST x Elt x Int -> LIST  remove: LIST x Int -> LIST  get: LIST x Int -> Elt  find: LIST x Elt -> Int  size: LIST -> Nat  empty: LIST-> Bool  Signature of a list.  new()  Stack Axiom: pop(push(new(), -3)) = \*what\*?  push(new(), 7)  Stack Axiom: pop(push(push(new(), 7, 4))) = \*what\*?  0  Stack Axiom: size(new()) = \*what\*?  1  Stack Axiom: size(push(new(), 6)) = \*what\*?  -8  Stack Axiom: top(push(push(new(), 3) -8)) = \*what\*?  new()  Stack Axiom: pop(push(new(), -3)) = \*what\*?  2  Stack Axiom: top(pop(push(push(new(), 2), 7))) = \*what\*?  size(S) + 1  Stack Axiom: size(push(S, i)) = \*what\*?  new()  Stack Axiom: pop(new()) = \*what\*?  S  Stack Axiom: pop(push(S, i)) = \*what\*?  Error  Stack Axiom: top(new()) = \*what\*?  i  Stack Axiom: top(push(S, i)) = \*what\*?  S  Stack Axiom: pop(push(S, i)) = \*what\*?  push(push(new(), 6), 4)  Stack Axiom: push(pop(push(push(new(), 6), 3)), 4) = \*what\*?  First In First Out (FIFO)  A queue is what kind of data structure?  First In Last Out (FILO)  A stack is what kind of data structure?  0  Queue Axiom: size(new()) = \*what\*?  size(Q) + 1  Queue Axiom: size(enq(Q, i)) = \*what\*?  Error  Queue Axiom: front(new()) = \*what\*?  new()  Queue Axiom: deq(new()) = \*what\*?  0  List Axiom: size(new()) = \*what\*?  size(L) + 1  List Axiom: size(ins(L, e, i)) = \*what\*?  True  List Axiom: empty(new()) = \*what\*?  False  List Axiom: empty(ins(L, e, i) = \*what\*?  Error  List Axiom: get(new(), i) = \*what\*?  Error  List Axiom: find(new(), e) = \*what\*?  new()  List Axiom: rem(new(), i) = \*what\*?  O(N^2)  The BigO time complexity of a sorted insort for a linked list is what?  Cross product all canonical operators with non-canonical operations to fully test the data structure.  Guttag's Method for proving that an ADT acts as intended in the signature.  Recursion  An algorithm that calls itself directly or indirectly is known as this.  Recursion Base Case  The problem is so small the solution is easy with no recursive call.  Recursion Recursive Case  The problem must be solved by calling the function again with a recursive call, but somehow on a smaller subset of the problem.  True  (T/F) Any loop can be expressed as recursion.  True  (T/F) Any recursive call can be expressed as a loop.  Stack  A data structure that stores information about the active subroutines of a computer program  Heap  A pool of memory set aside for dynamic runtime allocation. There is no systematic organization of this memory.  Call Frame  A template for the dynamic memory needed for the function to execute. It's put onto the stack.  O(1)  Consider a queue implemented by linked list, and containing N elements. If we manage to implement the enqueue operation in O(1) time worst case, what is the "best" worst case time complexity we can get for the dequeue operation?  2^N  A full binary tree has this many nodes.  O(N^2)  Sorting a sequence of N numbers by adding them successively to a list (finding and putting them into the proper location by value) has what worst case time complexity?  The Heap  In Java, "new" calls get memory from this runtime memory allocation area.  Activation Record  Another name for Stack Frame.  True  (T/F) It's easier to blowout the runtime stack than use up the entire heap. The runtime stack is smaller.  Tree  To represent hierarchical relationship between elements, which data structure is suitable?  Tree  A list where list cells are allowed to have more than one child going in one direction through the list.  Root  The top element of a tree.  Node  A data element in a tree  Edge  The connection between nodes in a tree.  False  (T/F) An edge in a tree cannot have data associated with it.  False  (T/F) An edge in a tree must be only one-way.  True  (T/F) An edge in a tree expresses a parent-child relationship.  The number of edges in the path.  The length of a path in a tree.  0  The length of the path from a node to itself.  0  The root of a tree has a depth of \*this\*.  Depth  The length of the path from a node to the root of the tree.  Height  The length of the longest path from a node to any one of its leaves.  0  Every leaf in a tree has a height of \*this\*.  Height of the Root  Height of any given tree.  Arity  The maximum number of children a node has determines the \*this\* of the tree.  Depth-First Traversal  General tree traversal in which one travels down a path until a leaf is reached.  Breadth-First Traversal  General tree traversal in which one visits all the children of a given node before going deeper into the tree.  new: -> BST  insert: Elt ->  remove: Elt ->  findMin: -> Elt  findMax: -> Elt  contains: Elt -> Bool  get: Elt -> BST  val: -> Elt  size: -> Nat  empty: -> Bool  height: -> Nat  Objected oriented (Java) signature of a binary search tree.  O(N log N)  Average case time complexity for binary search tree sort.  O(N^2) (It's bubble sort.)  Worst case time complexity for binary search tree sort.  O(N^2) (It's bubble sort.)  Average case time complexity for list sort.  O(N^2) (It's bubble sort.)  Worst case time complexity for list sort.  True  (T/F) There can be more than one unique BST containing the same elements.  O(N)  Worst case time complexity of any traversal of a BST.  O(N)  Average case time complexity of any traversal of a BST.  2^N (So performance is O(2^N))  Recursive calls in a Fibonacci function.  True  (T/F) In-order traversal in a binary search tree returns the rank over of that set.  Height + 1  Minimum number of nodes in a BST.  2^(Height + 1) - 1  Maximum number of nodes in a BST.  True  (T/F) Problems with BST are defined by the total number of nodes N in the BST.  False (This is only true with BSTs.)  (T/F) Problems with any tree are defined by the total number of nodes N in the tree.  Highest Priority Order  A priority queue is what kind of data structure?  False  (T/F) A priority queue is a binary heap.  A completely filled binary tree, except for the last row of leaves, filled left to right.  Heap-Structure Property  Minimum element is at the root, every parent is less than or equal to its children, and every path in the tree is in non-decreasing order.  Heap-Order Property  enqueue: PQUE x Elt x Priority -> PQUE  dequeue: PQUE -> PQUE  front: PQUE -> Elt  size: PQUE -> Nat  empty: PQUE -> Bool  Signature of a priority queue.  True  (T/F) Every subtree of a binary heap is also a binary heap.  O(N log N)  Worst case time complexity for binary heap sort.  O(N log N) + O(N log N)  \*Explicit\* worst case time complexity for binary heap sort using a normal, iterative insert methodology.  O(N) + O(N log N)  \*Explicit\* worst case time complexity for binary heap sort using the build method.  O(log N)  Consider implementing a PrQUE with a BST that somehow guarantees average case performance on all operations (not worst case). What is the worst case time complexity of "enq" on such a PrQUE with N items in it?  O(N)  Consider implementing a priority queue with a normal Linked Cell List. What is the worst case time complexity of the "enq" operation for a PrQUE with N items in it if it is sorted?  O(1)  Consider implementing a priority queue with a normal Linked Cell List. What is the worst case time complexity of the "enq" operation for a PrQUE with N items in it if it is not sorted?  The inverse of the height of the tree multiplied by the number of nodes in the row, but the last (leaves) node is not included in swaps.  The maximum number of swaps when building a binary heap using the build method.  Ambiguous (O(N log N) if worded "best worst case")  (T/F) We have an array of N integers. We wish to use a min binary heap to sort these integers into ascending order. Doing this operation will have a worst case time complexity described by O(N^2).  No  (Yes/No) Consider a min binary heap that allows duplicates. Does the heap preserve insertion order? If we have 2 items with priority 3 (lets call them 3 and 3') is it possible for 3 to be inserted into the heap first, but to come out second (after 3')?  True  (T/f) In an AVL tree with N nodes, each access operation (insert, remove, contains, findMin, etc.) has worst case time complexity of O(log N).  False  (T/F) Consider an AVL tree of N nodes. Consider the paths from root to leaves. The longest such path and the shortest such path will always differ by no more than 1.  O(log N)  Worst case time complexity of binary search in a sorted list.  True  (T/F) Any n arity tree can be represented as a binary tree.  False (The first half is False, a recursive function can have as many base cases as necessary. The second half is True. A recursive function must have at least one of each though.)  (T/F) A recursive function must have one base case, but can have as many recursive cases as necessary.  First In Last Out (FILO)  The behavior discipline of the runtime stack.  True  (T/F) Arrays in Java are covariant.  False  (T/F) Generics in Java are covariant.  Algorithm  A clearly specified set of simple instructions to be followed to solve a problem.  Upper  Big-O notation , O(), is \*this\* kind of bound.  Lower  Big-Omega notation, 𝛀(), is \*this\* kind of bound.  Equal  Big-Theta notation, 𝚯(), is \*this\* kind of bound.  Strong-Upper (Does not allow for equivalence)  Little-O notation, o(), is \*this\* kind of bound.  Online Algorithm  In this type of algorithm at any point in time the algorithm can correctly give an answer to the subsequence problem for the data it has already read.  True  (T/F) A static nested class interacts with the instance members of its outer class (and other classes) just like any other top-level class. In effect, a static nested class is behaviorally a top-level class that has been nested in another top-level class for packaging convenience.  True  (T/F) A nested class in Java has access to private fields of its top-level parent.  Sentinel Node  An extra node in a LinkedList that is tracked solely for the sake of making fewer edge cases in insert/remove operations.  Header Node  A sentinel node in a LinknedList at the head of the list.  Tail Node  A sentinel node in a LinkedList at the tail (end) of the list.  Reverse Polish Notation (Postfix Notation)  A mathematical notation in which operators follow their operand. It is easily implemented by a stack, and so allows for O(N) analysis of mathematical expressions.  2^(h + 1) - 1 - (h + 1)  The sum of the heights of all of the nodes in a perfect binary heap (or tree) containing 2^(h + 1) nodes.  d-Heap  A generalized heap in which every node has d number of children. A 4-Heap seems to be the most efficient if inserts and deletes are expected to be equal.  Null Path Length (npl(X))  \*This\* of any node X is the length of the shortest path from X to a node without two children.  Leftist Heap Property  For every node X in a heap, the null path length of the left child is at least as large as that of the right child.  2^(r) - 1  A leftist tree with r nodes on the right path must have at least \*this\* many nodes.  O(log N)  The best worst case time complexity of merge for a leftist heap.  Skew Heap  A binary tree obeying heap-order property without having any structural property; no information is contained about the null path length of any node.  True  (T/F) During a merge, skew heaps always swap elements.  Binomial Queue  A collection of heap-ordered trees, but is not heap-ordered itself.  Forest  A collection of heap ordered trees.  Circular Array  This implementation of a queue via an array allows for O(1) for both enque and deque operations. It uses looping to avoid moving elements along the array in O(N) time.  log(N)  In a BST of N nodes, this is the minimum height.  O(N) (THIS IS RIGHT. DON'T CHANGE IT.)  BigO heapsort average case time complexity.  O(N^2)  BigO insertion sort worst case time complexity.  O(N^2)  BigO insertion sort average case time complexity.  O(N^2)  BigO bubble sort worst case time complexity.  O(N^2)  BigO bubble sort average case time complexity. |  |  |

CST8130: Data Structures Midterm #1 Winter 2019   
- 3 -   
Part 1 – Multiple Choice (12 Questions – 1 mark each) Answer on this paper.   
1. What would be the MOST correct code to initialize the array and instantiate all of its elements in the   
initial constructor of the CST8130 class?   
a) students= new Student[max];   
for (int i=0; i<max; i++)   
students[i] = new Student();   
  
b) if (max < 0)   
students = new Student[10];   
else students = new Student[max];   
for (int i=0; i<max; i++)   
students[i] = new Student();   
  
c) numStudents = 0;   
students = new Student[10];   
  
d) if (max < 0)   
students = new Student[10];   
else students = new Student[max];   
for (int i=0; i<students.length; i++)   
students[i] = new Student();   
  
2. If a reference, an int and a float each use 1 byte in memory; and a String object uses 2 bytes in   
memory, how many bytes of memory in total is used by students with this declaration?   
Student [ ] students = new Student[6];   
  
a) 1 b) 5 c) 6 d) 7 e) 36 f) 37   
  
3. If a reference, an int and a float each use 1 byte in memory; and a String object uses 2 bytes in   
memory, how many bytes of memory in total is used by newOne with this declaration?   
Student newOne = new Student();   
  
a) 1 b) 4 c) 5 d)6 e) 7 f) 8   
  
4. What is the BigO of the following code segment? Assume method doIt() is O(n).   
j=n;   
while (j>0) { a) O(n2logn) b) O(n2)   
doIt(); c) O(nlogn) d) O(logn)   
doIt(); e) none of the other answers   
j-=2;   
}   
  
5. Of the following, which is the BEST algorithm measurement?   
a) O(n2) b) O(nlogn) c) O(n2 log n) d) O(1)   
  
6. Of the following, which is the WORST algorithm measurement?   
O(n2) b) O(nlogn) c) O(n^2 log n) d) O(1)   
  
7. Assuming the following code has executed: Student me = new Student();   
me.addFromKeyboard(keyboard);   
with 12345 entered as studentNumber, “Linda Crane” as studentName and 2.6 as studentGPA.   
  
What will display from: if (me.isEligibleForCoop())   
System.out.println (me);   
else System.out.println (“not eligible”);   
  
a) me c) not eligible   
b) Linda Crane 12345 has average 2.6 d) none of the other answers

CST8130: Data Structures Midterm #1 Winter 2019   
- 4 -   
8. Given the method: public static int recurse(int x) {   
if (x<2)   
return x;   
else return (x+ recurse(x-1));   
}   
What would display from System.out.println (recurse(3)); ?   
  
a) 1 b) 3 c) 5 d) 6 e) none of other answers   
  
9. Given the method: public static int doThis (int num) {   
int total = 0;   
for (int i=0; i<num; i++)   
for (int j=0; j<num-i; j++)   
total ++;   
return total;   
}   
What will display from System.out.println (doThis(4));   
  
a) 7 b) 8 c) 9 d) 10 e) 11   
  
10. What is the value of y after the following statements execute?   
int x= 5;   
double y = 4.2 + 3 \* x / 2;   
a) 10.2 b) 11.2 c) 11.7 d) 18.0 e) 31.5   
  
11. Student student = new Student(); What value is in studentNumber of student after this executes?   
a) 0 b) 111111 c) 123456   
  
12. From Assignment 1....Given the Packet class with method processNotFound(); and RoutingPacket   
class (which extends Packet) which also has method processNotFound();   
  
Packet newOne = new RoutingPacket();   
newOne.processNotFound();   
  
Which version of processNotFound() will Java execute?   
a) The one in Packet class   
b) The one in RoutingPacketclass   
c) The one in Packet class and then the one in RoutingPacket class   
d) The one in RoutingPacket class and then the one in Packet class   
  
Part 2 – Short Answer (15 marks)   
13. Two of the sorting algorithms we studied were QuickSort and BubbleSort. What property of these   
algorithms might cause us to choose using one over the other? Explain completely. (2 marks)   
  
  
Quicksort has O(nlogn) which is faster than Bubblesort which is O(n2). So QuickSort is faster.

CST8130: Data Structures Midterm #1 Winter 2019   
- 5 -   
14. Like we did in Hybrid Activity 2, write in the box what this code will display. (4 marks)   
public class Test {   
public static void main(String[] args) {   
Stuff myStuff = new Stuff(15);   
System.out.println(myStuff);   
String s = "changed";   
myStuff.ChangeString(s);   
System.out.println(s);   
System.out.println(myStuff);   
  
}   
}   
  
public class Stuff {   
private int n = 4;   
private String s = "Original";   
  
public Stuff(int n) {   
this.n = n / 6;   
}   
  
public void ChangeString(String s) {   
this.s = s;   
s = "changed again";   
System.out.println(s);   
}   
  
public String toString() {   
return "Stuff has n = " + n + " and s = " + s;   
}   
}

Answer here:   
Stuff has n = 2 and s = Original   
changed again   
changed   
Stuff has n = 2 and s = changed  
  
15. Draw a memory map of ourClass object of type CST8130 declared below. Assume that code to add   
two students to the array has been executed as well.   
CST8130 ourClass = new CST8130(5); (3 marks)

CST8130: Data Structures Midterm #1 Winter 2019   
- 6 -   
16. Write the Java code for a method in the CST8130 class with signature   
  
public boolean addStudent (Student toAddObj)   
  
This method should add the object toAddObj (which already has complete data in it) to the students array   
Make sure you handle every case (your code should never crash). (3 marks)   
  
  
  
if (numStudents < students.length) {   
Students[numStudents++] = toAddObj;   
return true;   
} else {   
System.out.println (“No room to add students);   
return false;   
  
  
  
  
  
17. Assume that the code for all the methods listed in the Student and CST8130 classes has been written.   
Write the code in method main that would add 5 students to a CST8130 object. (3 marks)   
public static void main (String [] args) {   
  
Scanner keyboard = new Scanner (System.in);   
CST8130 ourClass = new CST8130(5);   
  
// write rest of code here   
  
Student student = null;   
for (int i=0; i<5; i++) {   
student = new Student();   
if (student.addFromKeyboard(keyboard)) {   
ourClass.addStudent (student);   
} else {   
System.out.println (“Invalid add from keyboard...reenter”);   
i--;   
}