|  |  |  |
| --- | --- | --- |
| **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.**

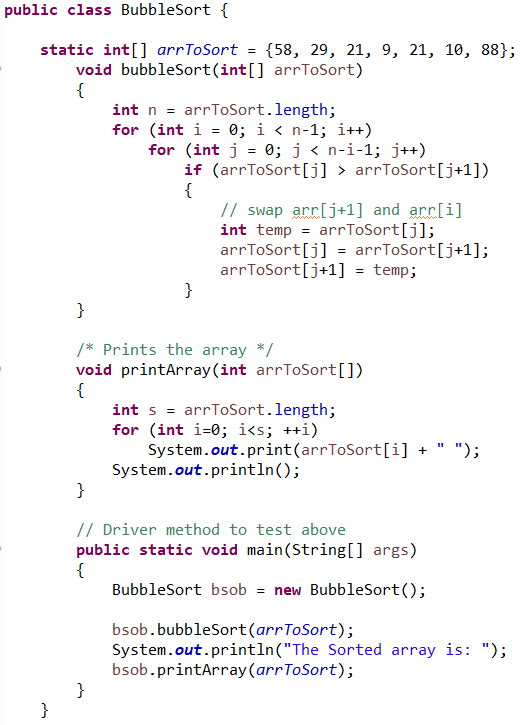
|  |  |  |
| --- | --- | --- |
| **Question 3** |  | 0 / 1 point |

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)** |

|  |  |  |
| --- | --- | --- |
| **uestion 1** |  | 0 / 1 point |

What is the BigO for this algorithm where the BigO for method doSomething is O(n)?

for (int i =0; i < n; i++) {  
      doSomething ();  
}

|  |  |  |  |
| --- | --- | --- | --- |
| Incorrect Response | O(n) | | |
|  | O(nlogn) | | |
| Correct Answer | O(n2) | | |
|  | O(n3) | | |
| **Question 2** | |  | 1 / 1 point | |

What displays from:

public static int recurse (int n) {  
     if (n > 0)   
          return (n\* recurse(n-1));  
  else    
 return 1;  
}  
  
public static void main (String [] argS) {        
 System.out.println (recurse (4));  
}

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | 6 |
|  | | | 24 |
|  | | | 12 |
|  | | | 1 |
|  |  |
|  | |

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

The poll() method retrieves and removes the head of a queue, or returns null if the queue is empty.

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

What is the algorithm efficiency for insertion to the last open element in an array (assuming there is an empty space)?

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

What is the algorithm efficiency for the .add(E) method in the ArrayList class?  (Assume there is capacity in the array)

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

What is the algorithm efficiency of the .add(index, E) method in the ArrayList class?

|  |  |  |  |
| --- | --- | --- | --- |
|  | O(1) | | |
|  | O(logn) | | |
|  | O(n) | | |
|  | O(nlogn) | | |
| **Question 5** | |  | 1 / 1 point | |

What is the BEST algorithm efficiency for search in a sorted array?

|  |  |  |  |
| --- | --- | --- | --- |
|  | O(1) | | |
|  | O(logn) | | |
|  | O(n) | | |
|  | O(nlogn) | | |
| **Question 6** | |  | 1 / 1 point | |

What is the algorithm efficiency of .get(index) in the ArrayList class?

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

Please copy your delete code from the LList class

//Look for it in the notes, study it and paste it in the answer box

public LLNode delete (String oneToDelete) {  
LLNode removedOne = null;  
LLNode temp = head;  
LLNode prev = head;  
while (temp != null) {  
if (temp.toString().equals(oneToDelete)) {  
removedOne = temp;  
if (temp ==  
head)   
head = temp.getNext();  
else   
prev.updateNode(temp.getNext());  
break;  
}  
else {   
prev = temp;  
temp = temp.getNext();  
}  
} // end while  
return removedOne;  
}

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

Add 1 item?  
Singly linked list with head (next ref only): O(

\_\_\_1\_\_\_(33.33 %)

)  
Singly linked list with head and tail (next ref only): O(

\_\_\_1\_\_\_(33.33 %)

)  
Doubly linked list – head and tail (next and prev references): O(

\_\_\_1\_\_\_(33.33 %)

)

|  |
| --- |
| [View Feedback](javascript://) |
| **Question 2** | |  | 1 / 1 point |

Add 1 item to unsorted structure?  
Singly linked list with head (next ref only): O(

\_\_\_1\_\_\_(33.33 %)

)  
Singly linked list with head and tail (next ref only): O(

\_\_\_1\_\_\_(33.33 %)

)  
Doubly linked list – head and tail (next and prev references): O(

\_\_\_1\_\_\_(33.33 %)

)

|  |
| --- |
| [View Feedback](javascript://) |
| **Question 3** | |  | 1 / 1 point |

Remove 1 item at beginning/head?  
Singly linked list with head (next ref only): O(

\_\_\_1\_\_\_(33.33 %)

)  
Singly linked list with head and tail (next ref only): O(

\_\_\_1\_\_\_(33.33 %)

)  
Doubly linked list – head and tail (next and prev references): O(

\_\_\_1\_\_\_(33.33 %)

)

|  |
| --- |
| [View Feedback](javascript://) |
| **Question 4** | |  | 1 / 1 point |

Remove 1 item at end/tail?  
Singly linked list with head (next ref only): O(

\_\_\_n\_\_\_(33.33 %)

)  
Singly linked list with head and tail (next ref only): O(

\_\_\_n\_\_\_(33.33 %)

)  
Doubly linked list – head and tail (next and prev references): O(

\_\_\_1\_\_\_(33.33 %)

)

|  |
| --- |
| [View Feedback](javascript://) |
| **Question 5** | |  | 1 / 1 point |

Remove specific item in middle?  
Singly linked list with head (next ref only): O(

\_\_\_n\_\_\_(33.33 %)

)  
Singly linked list with head and tail (next ref only): O(

\_\_\_n\_\_\_(33.33 %)

)  
Doubly linked list – head and tail (next and prev references): O(

\_\_\_n\_\_\_(33.33 %)

)

|  |
| --- |
| [View Feedback](javascript://) |
| **Question 6** | |  | 1 / 1 point |

time complexity of Search ?  
Singly linked list with head (next ref only): O(

\_\_\_n\_\_\_(33.33 %)

)  
Singly linked list with head and tail (next ref only): O(

\_\_\_n\_\_\_(33.33 %)

)  
Doubly linked list – head and tail (next and prev references): O(

\_\_\_n\_\_\_(33.33 %)

)

|  |
| --- |
| [View Feedback](javascript://) |
| Memory Efficiency | |

Answer these questions with an explanation.  For example, the question "How much memory would you need for an array of 10 ints?" should be answered:

- One reference for the array  
- 10 references for each element in the array  
- 10 times the size of an int

|  |  |  |
| --- | --- | --- |
| **Question 7** |  | 1 / 1 point |

Explain how much memory is required for (n) items in a singly linked list with head (next ref only) of data objects?

- One reference for the singly linked array  
-2 references (one for object, one for next)  
- n times the size of object

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

|  |
| --- |
| [View Feedback](javascript://) |
| **Question 8** | |  | 1 / 1 point |

Explain how much memory is required for (n) items in a singly linked list with head and tail (next ref only) of data objects?

- One reference for head and one reference for tail  
-2 references (one for object, one for next)  
- n times the size of object

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

|  |
| --- |
| [View Feedback](javascript://) |
| **Question 9** | |  | 1 / 1 point |

Explain how much memory is required for (n) items in a doubly linked list – head and tail (next and prev references) of data objects?

- One reference for head and one reference for tail  
-3 references (one for object, one for next and one for previous)  
- n times the size of object

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

|  |
| --- |
| [View Feedback](javascript://) |

Written: Nov 12, 2021 9:22 AM - Nov 12, 2021 9:45 AM

**Submission View**

Your quiz has been submitted successfully.

|  |
| --- |
| 53, 60, 22, 54, 10, 24, 23 |

The following questions will be based on the Binary Search Tree resulting from adding the data provided in this order: 53, 60, 22, 54, 10, 24, 23

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

Draw the binary tree.

You should write it something like this

  A  
 /  \  
 B  C  
/ \  /  \  
 D E

                      53  
           22                 60  
     10        24      54  
             23

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

|  |
| --- |
| [View Feedback](javascript://) |
| **Question 2** | |  | 1 / 1 point |

Indicate the in-order traversal display for the above Binary Tree?

|  |  |
| --- | --- |
| Answer: | 10 22 23 24 53 54 60 |

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

Indicate the pre-order traversal display for the above Binary Tree?

|  |  |
| --- | --- |
| Answer: | 53 22 10 24 23 60 54 |

|  |  |  |
| --- | --- | --- |
| **Question 4** |  | 1 / 1 point |

What is the height of the binary tree?

|  |  |
| --- | --- |
| Answer: | 4 |

|  |  |  |
| --- | --- | --- |
| **Question 5** |  | 1 / 1 point |

Draw the binary tree.

You should write it something like this

  A  
 /  \  
 B  C  
/ \  /  \  
 D E

                               4  
                       1            10  
                                             16  
                                      14          22  
                                             15

|  |  |  |
| --- | --- | --- |
| **Question 6** |  | 1 / 1 point |

Indicate the in-order traversal display for the above Binary Tree?

|  |  |
| --- | --- |
| Answer: | 1 4 10 14 15 16 22 |
| |  |  |  | | --- | --- | --- | | **Question 7** |  | 0 / 1 point |   Indicate the post-order traversal display for the above Binary Tree?   |  |  |  | | --- | --- | --- | | Answer: | 4 1 10 16 14 15 22 | Incorrect Response**(/1, 15, 14, 22, 16, 10, 4/, /1,15,14,22,16,10,4/, /1.\*15.\*14.\*22.\*16.\*10.\*4/)** |  |  |  |  | | --- | --- | --- | | **Question 8** |  | 1 / 1 point |   What is the height of the binary tree?   |  |  | | --- | --- | | Answer: | 5 |  |  |  |  | | --- | --- | --- | | **Question 9** |  | 0 / 1 point |   For the following numbers – what order do the numbers need to be entered in to keep this a complete tree after every step.  10, 22, 23, 24, 30, 54, 60   |  |  |  | | --- | --- | --- | | Answer: | post order | Incorrect Response**(/24, 22, 54, 10, 23, 30, 60/, /24,22,54,10,23,30,60/, /24.\*22.\*54.\*10.\*23.\*30.\*60/)** |  |  |  |  | | --- | --- | --- | | **Question 1** |  | 1 / 1 point |   A set can contain duplicate values as it is just a sequence of values.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  | True | | | |  |  | False | | | | **Question 2** | | |  | 1 / 1 point | |   Map can contain duplicate keys which can point to different values.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  | True | | | |  |  | False | | | | **Question 3** | | |  | 1 / 1 point | |   The best performing Map for search operation is   |  |  |  |  | | --- | --- | --- | --- | |  | HashMap as it has **a hashtable-based implementation** | | | |  | Tree Map as it stores it elements in natural order | | | |  | LinkHashMap as it stores order in which keys are inserted into it | | | | **Question 4** | |  | 0 / 3 points | |   Write a program that takes the numbers 145623 to 145723 and displays the results in a table :   * Display the number * Display the last two digits of the number * Display the middle two digits of the number * Add each of the digits together and display the last two digits of the sum   Example:  Number           last two digits               middle two digits          last digits of sum  145623        23         56       21  145624        24         56       22    Copy your code below.  public class hash {     public static int sumNumbers(int s) {         int sum = 0;         while (s != 0) {                 sum += s % 10;                 s = s / 10;         }         return sum;  }               public static void main(String[] args) {         int numbers[] = {145623, 145723};          System.out.printf("%-20s%-20s%-20s%-20s\n",                     "number", "last two digits",                         "middle two digits", "last digits of sum");          for(int number: numbers) {                 System.out.printf("%-20d%-20d%-20d%-20d\n",                         number, number % 100, (number / 100) % 100, sumNumbers(number) % 100);         }             } }  **public** **class** HashAlgoEx {  **public** **static** **void** main(String[] args) {  System.***out***.println ("For numbers 145623 to 145624");  System.***out***.println ("Number           last two digits               middle two digits          last digits of sum");  **int** lastTwo;  **int** middleTwo;  **int** computedTwo;  **int** [] lastTwoArray = **new** **int**[100];  **int** [] midTwoArray = **new** **int**[100];  **int** [] sumArray = **new** **int**[100];  **for** (**int** i=145623; i < 145724; i++ ) {  lastTwo = i%100;  middleTwo = i/100%100;  computedTwo = ((i/100000) + (i/10000%10) + (i/1000%10) + (i/100%10) + (i/10%10) + (i%10))%100;  System.***out***.println (i +"        " + lastTwo + "         " +middleTwo + "       " + computedTwo);  }  System.***out***.println ("For 100 random numbers");  Random random = **new** Random();  System.***out***.println ("\n\nNumber           last two digits               middle two digits          last digits of sum");  **for** (**int** j=0; j < 100; j++ ) {  **int** i = random.nextInt(1000000) + 1;  lastTwo = i%100;  lastTwoArray[lastTwo]++;  middleTwo = i/100%100;  midTwoArray[middleTwo]++;  computedTwo = ((i/100000) + (i/10000%10) + (i/1000%10) + (i/100%10) + (i/10%10) + (i%10))%100;  sumArray[computedTwo]++;  System.***out***.println (i +"        " + lastTwo + "         " +middleTwo + "       " + computedTwo);  }  System.***out***.println("Distribution of array indexes");  **for** (**int** i = 0; i < 100; i++) {  System.***out***.println (i +"        " + lastTwoArray[i] + "         " +midTwoArray[i] + "       " + sumArray[i]);  }  Which statements are correct?   |  |  | | --- | --- | |  | Hashing is the process of converting a given key into another value. | |  | A good hash function uses a **two-way** hashing algorithm, or in other words, the hash can be converted back into the original key. | |  | A **hash function** is used to generate the new value according to a mathematical algorithm. | |  | The result of a hash function is known as a **hash value** or simply, a **hash**. | |  | **Hash Collision** is a phenomenon when two keys can generate the same hash. | |  |