|  |  |  |
| --- | --- | --- |
| **APCS Exposure Java** | **Exercises 10.01-05** | **Date:** |
| **Name: KEY** | | **Period:** |

1. How many values can be stored by a simple data type?

*1*

2. What is a *data structure*?

*A data structure is a data type whose components are smaller data structures and/or simple data types.*

3. Any data type that can store more than one value is a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

*data structure*

4. What is another word for an *array*?

*A subscripted variable*

5. *Arrays* first became popular with what programming language?

*FORTRAN*

6. What is an *array*?

*An array is a data structure with one, or more, elements of the same type.*

7. A one-dimensional array is frequently also called a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

*vector*

8. A two-dimensional array is frequently also called a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

*matrix*

9. If you need to store several pieces of data for many people, and you create one array for their names, another array for their addresses, another for their birthdates, another for their social security numbers, etc. What kind of arrays have you created?

*Parallel Arrays*

10. *Records* first became popular with what programming language?

*COBOL*

11. What is a *record*?

*A record is a data structure with one, or more, elements, called fields, of the same or different data types.*

12. What is a *file*?

*A file is an internal data structure - with an unspecified number of elements of the same type - assigned to an external file name.*

13. What does the *file* data structure allow?

*The file data structure allows transfer of data between internal and external storage.*

14. What is a stack?

*A stack is a data structure with elements of the same type. Data elements of the stack data structure can only be accessed (stored or retrieved) at one end of the stack in a LIFO manner.*

15. What does *LIFO* stand for?

*Last In, First Out*

16. What is the *Improved Data Structure Definition*?

*A data structure is a data type whose components are smaller data structures and/or simple data types.*

*The storing and retrieval of the data elements is performed by accessing methods that characterize the data structure.*

17. What is the *Improved Array Definition*?

*An array is a data structure with a fixed number of elements of the same type.*

*Every element of the array can be accessed directly.*

18. What do an *array* and a *stack* have in common?

*They both store a fixed number of elements of the same type.*

19. Refer to the previous question.

If they have this in common, why are they considered different data structures?

*They access the information differently.*

20. What is meant by a *contiguous* block of memory?

*a solid block from memory so that each array element is adjacent to the next array element in memory*

21. Write the Java code to declare an **int** array called *ages*?

*int ages[];*

22. Refer to the previous question. Write the Java code that will allocate memory for 30 *ages*?

*ages = new int[30];*

23. Java arrays indicate individual elements with an \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ inside two brackets.

*index*

24. What data type will the array index always be?

*int*

25. What is the first array index?

*0*

26. Write the Java code that will both declare and allocate memory for an array of 40 **String**s called *names* in one single programming statement.

*String names[] = new String[40];*

27. If the individual elements of an **int** array are not initialized, what is stored in them by default?

*0*

28. Write the Java code that will both declare and allocate memory for an initialized array of **double** values called **bob**. The initialized values will be **1.1**, **2.2**, **3.3**, **4.4**, and **5.5**.

*double bob[] = {1.1,2.2,3.3,4.4,5.5};*

29. Refer to the previous question. In this situation, does the size of the array need to be specified?

*No*

30. What does **length** do when used with a *Java Static Array*?

*It tells you how many elements are in the array.*

31. Refer to the previous question. Is **length** a method?

*No*

32. Can the **length** field be changed?

*No*

33. What is the difference between a *static array* and a *dynamic array*?

*You cannot change the size of a static array.*

*You can change the size of a dynamic array.*

34. When generating random integers, what is the purpose of the *seed*?

*The seed insures that every program execution produces an identical set of numbers.*

35. Look at program ***Java1008.java***. How would you change the code if you were using **Math.random**?

*Change the statement list[k] = random.nextInt(900) + 100; to*

*list[k] = (int) Math.floor(Math.random \* 900) + 100;*

36. Look at program ***Java1009.java***. This program displays random sentences. How is that possible?

*There are 4 different arrays. One has 7 ranks; one has 7 names; one has 7 actions and the last one has 7 locations. 4 random indexes are chosen, one for each array. This allows me to create a sentence with a random rank, followed by a random name, followed by a random action followed by a random location.*

37. Whether you have an array of integers, an array of real numbers, or an array of strings, what data type will the *index* always be?

*int*

38. What is the difference between the *old* **for** loop and the *new enhanced* **for** loop?

*The enhanced for loop can manage access to array elements without using an index or a loop counter.*

39. Does the *new enhanced* **for** loop only work with **String** objects?

*No, it works with many data structures.*

40. Does the *new enhanced* **for** loop only work with *primitive* data types?

*No, an object is required.*

41. Look at programs ***Java1011.java*** and ***Java1012.java***. How was the second program changed?

Why does this still work?

*The array element identifier was changed from a String to an Object.*

*This still works because Strings are Objects.*

42. The *enhanced* **for** loop is called the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ loop.

*for..each*

43. Does the *enhanced* **for** loop replace the older **for** loop? Explain why.

*No, because it is not possible to access specific array elements.*