Capitulo 8

This activity contains 32 questions.

1. *Section 8.2 Pointer Variable Declarations and Initialization*

8.2 Q1: Pointers cannot be used to:

- ○ *Reference values directly.*
- ○ *Manipulate dynamic data structures.*
- ○ *Contain memory addresses.*
- ○ *Pass an argument by reference.*

2. *8.2 Q2: Pointers may be assigned to which of the following?*

- ○ *Any integer values.*
- ○ *NULL.*
- ○ *An address.*
- ○ *Both (b) and (c).*

3. *8.2 Q3: What does the following declaration declare?*

*int *countPtr, count;*

- ○ *Two int variables.*
- ○ *Two pointers to ints.*
- ○ *One pointer to an int and one int variable.*
- ○ *The declaration is invalid.*

4. *Section 8.3 Pointer Operators*

8.3 Q1: The & operator can be applied to:

- ○ *constants.*
- ○ *rvalues.*
- ○ *lvalues.*
- ○ *string literals.*

5. *8.3 Q2: All of the following could cause a fatal execution-time error except:*

- ○ *Dereferencing a pointer that has not been initialized properly.*
- ○ *Dereferencing a pointer that has not been assigned to point to a specific address.*
- ○ *Dereferencing a variable that is not a pointer.*
- ○ *Dereferencing a null pointer.*

6. *8.3 Q3: Three of the following expressions have the same value. Which of the following expressions has a value different from the others'?*

- ○ **Ptr.*
- ○ *Ptr.*
- ○ *&*Ptr.*
- ○ **&Ptr.*

7. *Section 8.4 Passing Arguments to Functions by Reference with Pointers*

8.4 Q1: Which of the following is not a valid way to pass arguments to a function in C++?

- ○ *By value.*
- ○ *By value with pointer arguments.*
- ○ *By reference with reference arguments.*
- ○ *By reference with pointer arguments.*

8. *8.4 Q2: When a compiler encounters a function parameter for a single-subscripted array of the form int a[], it converts the parameter to:*

- ○ *int * a.*
- ○ *No conversion is necessary.*
- ○ *int a.*
- ○ *int &a.*

9. *Section 8.5 Using const with Pointers*

8.5 Q1: A function that modifies an array by using pointer arithmetic such as ++ptr to process every value should have a parameter that is:

- ○ *A constant pointer to nonconstant data.*
- ○ *A nonconstant pointer to constant data.*
- ○ *A constant pointer to constant data.*
- ○ *A nonconstant pointer to nonconstant data.*

10. *8.5 Q2: A function that prints a string by using pointer arithmetic such as ++ptr to output each character should have a parameter that is:*

- ○ *A constant pointer to nonconstant data.*
- ○ *A nonconstant pointer to nonconstant data.*
- ○ *A constant pointer to constant data.*
- ○ *A nonconstant pointer to constant data.*

11. *8.5 Q3: An array name is:*

- ○ *A constant pointer to constant data.*
- ○ *A constant pointer to nonconstant data.*
- ○ *A nonconstant pointer to constant data.*
- ○ *A nonconstant pointer to nonconstant data.*

12. *8.5 Q4: What method should be used to pass an array to a function that does not modify the array and only looks at it using array subscript notation:*

- ○ *A constant pointer to constant data.*
- ○ *A nonconstant pointer to constant data.*
- ○ *A constant pointer to nonconstant data.*
- ○ *A nonconstant pointer to nonconstant data.*

13. *Section 8.6 Selection Sort Using Pass-by-Reference*

8.6 Q1: After the ith iteration of the selection sort:

- ○ *None of the above.*
- ○ *The smallest i items of the array will be sorted into decreasing order in the first i elements of the array.*
- ○ *The largest i items of the array will be sorted into decreasing order in the last i elements of the array.*

- ○ *The smallest i items of the array will be sorted into increasing order in the first i elements of the array.*

14. *8.6 Q2: To follow the principle of least privilege, the selectionSort function should receive the array to be sorted as:*

- ○ *A nonconstant pointer to constant data.*
- ○ *A constant pointer to constant data.*
- ○ *A constant pointer to nonconstant data.*
- ○ *A nonconstant pointer to nonconstant data.*

15. *Section 8.7 sizeof Operators*

8.7 Q1: sizeof:

- ○ *Returns the total number of elements in an array.*
- ○ *Is a binary operator.*
- ○ *Usually returns a double.*
- ○ *Returns the total number of bytes in a variable.*

16. *8.7 Q2: Which of the following gives the number of elements in the int array r[ ]?*

- ○ *sizeof (*r).*
- ○ *sizeof r.*
- ○ *sizeof (*r) / sizeof (int).*
- ○ *sizeof r / sizeof (int).*

17. *Section 8.8 Pointer Expressions and Pointer Arithmetic*

8.8 Q1: Which of the following can have a pointer as an operand?

- ○ **=.*
- ○ *++.*
- ○ */.*
- ○ *%.*

18. *8.8 Q2: Given that k is an integer array starting at location 2000, kPtr is a pointer to k and each integer is stored in 4 bytes of memory, what location does kPtr + 3 point to?*

- ◯ *2012.*
- ◯ *2024.*
- ◯ *2003.*
- ◯ *2006.*

19. *8.8 Q3: A pointer can not be assigned to:*

- ◯ *Another pointer of the same type without using the cast operator.*
- ◯ *Any other pointer by using the cast operator.*
- ◯ *A pointer to void without using the cast operator.*
- ◯ *A pointer of a type other than its own type and void without using the cast operator.*

20. *8.8 Q4: Comparing pointers and performing pointer arithmetic on them is meaningless unless:*

- ◯ *They point to elements of the same array.*
- ◯ *You are trying to compare and perform pointer arithmetic on the values to which they point.*
- ◯ *They point to arrays of equal size.*
- ◯ *They point to arrays of the same type.*

21. *Section 8.9 Relationship Between Pointers and Arrays*

8.9 Q1: Assuming that t is an array and tPtr is a pointer to that array, which expression refers to the address of the fourth element?

- ◯ *&t[3].*
- ◯ **(tPtr + 3).*
- ◯ **(t + 3).*
- ◯ *tPtr[3].*

22. *8.9 Q2: Consider the following function:*

```
void reverse( char * string1, const char * string2 )
{
   int stringsize = sizeof( string1 )/sizeof( char );
   *( string1 + stringsize -1 ) = '\0';
   string1 = string1 + stringsize - 2;
   for ( ; *string2 != '\0'; string1--, string2++ )
      *string1 = *string2;
}
```

What method does the function use to refer to array elements?

- ◯ *Pointer subscript notation.*
- ◯ *Array subscript notation.*
- ◯ *Pointer/offset notation where the pointer is actually the name of the array.*
- ◯ *Pointer/offset notation.*

23.

Section 8.10 Arrays of Pointers

8.10 Q1: A string array:

- ◯ *Is always less memory efficient than an equivalent double-subscripted array.*
- ◯ *Can only provide access to strings of a certain length.*
- ◯ *Stores an actual string in each of its elements.*
- ◯ *Is actually an array of pointers.*

24.

8.10 Q2: A string array is commonly used for:

- ◯ *Storing an extremely long string.*
- ◯ *Command-line arguments.*
- ◯ *Storing multiple copies of the same string.*
- ◯ *Displaying floating-point numbers to the screen.*

25.

Section 8.11 Case Study: Card Shuffling and Dealing Simulation

8.11 Q1: An algorithm that could execute for an unknown amount of time because it depends on random numbers may:

- ◯ *Get caught in an infinite loop.*
- ◯ *Have a redundancy.*
- ◯ *Issue a compiler error.*
- ◯ *Suffer from indefinite postponement.*

26. *Section 8.12 Function Pointers*

8.12 Q1: Which of the following is not true of pointers to functions?

- ○ *They can not be assigned to other function pointers.*
- ○ *They contain the starting address of the function code.*
- ○ *They can be stored in arrays.*
- ○ *They are dereferenced in order to call the function.*

27. *8.12 Q2: (*max)(num1, num2, num3);:*

- ○ *Is a call to the function pointed to by max.*
- ○ *Is the header for function max.*
- ○ *Is a declaration of a pointer to a function called max.*
- ○ *Is the prototype for function max.*

28. *Section 8.13 Introduction to Pointer-Based String Processing*

Section 8.13.1 Fundamentals of Characters and Pointer-Based String

8.13.1 Q1: Which of the following is not true?

- ○ *A string in C++ is an array of characters ending in the null character ('\0').*
- ○ *String literals are written inside of single quotes.*
- ○ *A string may be assigned in a declaration to either a character array or a variable of type char *.*
- ○ *A string may include letters, digits and various special characters (i.e., +, -, *).*

29. *8.13.1 Q2: cin.getline(superstring, 30); is equivalent to which of the following?*

- ○ *cin.getline(superstring, 30, '\0');.*
- ○ *cin.getline(superstring, 30, '\s');.*
- ○ *cin.getline(superstring, 30, '\n');.*
- ○ *cin.getline(superstring, 30, '\t');.*

30. *Section 8.13.2 String Manipulation Functions of the String-Handling Library*

8.13.2 Q1: Which of the following correctly copies the contents of string2 into string1? Assume that string2 is equal to "goodbye" and string1 is equal to "good morning"?

- ○ strcpy(string1, string2, 6);.
- ○ Strncpy(string1, string2, 5);.
- ○ strcpy(string1, string2);.
- ○ strncpy(string1, string2, 6);.

31. 8.13.2 Q2: Assuming that string1 = "hello" and string2 = "hello world", which of the following returns 0?

- ○ Strncmp(string1, string2, 5);.
- ○ strcmp(string1, string2, 6);.
- ○ strcmp(string1, string2);.
- ○ strncmp(string1, string2, 6);.

32. 8.13.2 Q3: strtok does not:

- ○ Completely tokenize the string the first time it is called.
- ○ Replace each delimiting character with '\0'.
- ○ Return a pointer to the token it creates.
- ○ Modify the input string.

Clear Answers / Start Over   Submit Answers for Grading

Answer choices in this exercise appear in a different order each time the page is loaded.