

Capítulo 22

This activity contains 27 questions.

1.

Section 22.1 Introduction

22.1 Q1: *The main difference between structures and classes is:*

- ☐ *How they access member variables.*
- ☐ *Whether they default to public or private access.*
- ☐ *That classes always require new be used with them while structures do not.*
- ☐ *There is no difference between structures and classes.*

2.

Section 22.2 Structure Definition

22.2 Q1: *Structure variable declarations can be incorporated into a structure definition by placing a comma-separated list of variable names:*

- ☐ *After the left brace and before the member declarations.*
- ☐ *After struct StructureName and before the left brace.*
- ☐ *After the right brace and before the semicolon.*
- ☐ *After the member declarations and before the right brace.*

3.

22.2 Q2: *Two structure variables of the same type with the same member values, when compared will:*

- ☐ *Never compare equally.*
- ☐ *Always compare equally.*
- ☐ *Result in a compile error.*
- ☐ *Sometimes compare equally.*

4.

Section 22.3 Initializing Structures

22.3 Q1: *Structure variables may not be initialized by:*

- ☐ *Assigning values to individual data members.*
- ☐ *Assigning user-defined values in the struct definition.*
- ☐ *Assigning the value of another structure variable of the same type.*
- ☐ *Array-like member initializer lists.*

5.

*Section 22.4 Using Structures with Functions**22.4 Q1: Arrays are:*

- ☐ *Passed by reference unless inside a structure.*
- ☐ *Always passed by value.*
- ☐ *Passed by reference unless their elements are structures.*
- ☐ *Always passed by reference.*

6.

*Section 22.5 typedef**22.5 Q1: typedef is used to:*

- ☐ *Cast one struct to another type.*
- ☐ *Make a struct private.*
- ☐ *Initialize struct members.*
- ☐ *Create a type name that is an alias for another type name.*

7.

22.5 Q2: The advantages of using typedef do not include:

- ☐ *Increasing the efficiency of accessing struct member variables.*
- ☐ *Making type names shorter.*
- ☐ *Making programs more portable by allowing data types to be easily changed to meet system specifications.*
- ☐ *Making programs more readable.*

8.

*Section 22.6 Example: High-Performance Card Shuffling and Dealing Simulation**22.6 Q1: Which is the proper way to create an array of structure variables of type Data?*

- ☐ *struct MyArray[Data];.*
- ☐ *Data MyArray[10];.*
- ☐ *Data struct myArray[10];.*
- ☐ *MyArray Data[10].*

9.

Section 22.7 Bitwise Operators

22.7 Q1: The most basic unit of data on a computer is the:

- ☐ Bit.
- ☐ int.
- ☐ File.
- ☐ Byte.

10.

22.7 Q2: Which of the following is not a bitwise operator?

- ☐ *.
- ☐ >>.
- ☐ ^.
- ☐ ~.

11.

22.7 Q3: Let $Bit1 = Bit2 = 1$. Which of the following does not have the same result as the others?

- ☐ $Bit1 \mid Bit2$.
- ☐ $Bit1 \wedge Bit2$.
- ☐ $Bit1 \& Bit2$.
- ☐ $\sim(\sim Bit2)$.

12.

22.7 Q4: Evaluate $(00001000 \& 11000101) \wedge (11110000)$.

- ☐ 11000000.
- ☐ 11110000.
- ☐ 00111101.
- ☐ 00001101.

13.

22.7 Q5: Let x be an unsigned int on a machine with 4-byte unsigned ints. What effect does

```
x>>=1 ;  
x<<=1 ;
```

have?

- ☐ There is no effect.
- ☐ The rightmost bit of x is set to 0.
- ☐ Both (b) and (c).
- ☐ The leftmost bit of x is set to 0.

14.

22.7 Q6: For any 8-bit x , which of the following does not result in zero?

- ☐ $x \wedge x$.
- ☐ $x \& (\sim x)$.
- ☐ $x \ll 8$.
- ☐ $x | x$.

15.

Section 22.8 Bit Fields

22.8 Q1: A bit field must be declared as a:

- ☐ *int* or *unsigned*.
- ☐ *char*.
- ☐ *float*.
- ☐ *long*.

16.

22.8 Q2: The number of bits in a bit field is specified with:

- ☐ A colon as in *bitfield : 4*.
- ☐ Brackets as in *bitfield[4]*.
- ☐ A dot as in *bitfield.4*.
- ☐ Parentheses as in *bitfield(4)*.

17.

22.8 Q3: _____ is not allowed.

- ☐ Having a bit field with a zero width.
- ☐ Padding a bit field with bits that cannot be accessed.
- ☐ Having an unnamed bit field.
- ☐ Accessing individual bits in a multi-bit bit field.

18.

Section 22.9 Character-Handling Library

22.9 Q1: The number 4 typically takes up _____ bit(s) when stored as a character on most of today's computers.

- ☐ Three.
- ☐ Eight.
- ☐ One.
- ☐ Four.

19.

22.9 Q2: The functions of the character-handling library typically manipulates characters as:

- ☐ floats.
- ☐ longs.
- ☐ ints.
- ☐ chars.

20.

22.9 Q3: The *isxdigit* function would return false on:

- ☐ g.
- ☐ a.
- ☐ 2.
- ☐ A.

21.

Section 22.10 Pointer-Based String-Conversion Functions

22.10 Q1: The *strtol* and *stroul* functions do not:

- ☐ Have the same return types.
- ☐ Have the ability to return data in base 8.
- ☐ Need a header file to be used.
- ☐ Take three arguments.

22.

22.10 Q2: The main difference between the functions *atof*, *atoi* and *atol* is:

- ☐ Their arguments.
- ☐ Their efficiency.
- ☐ Their header files.
- ☐ Their return types.



23.

Section 22.11 Search Functions of the Pointer-Based String-Handling Library

22.11 Q1: Which function would be the most useful for determining if a certain word is contained in a string representing a sentence?

- ☐ *strrchr.*
- ☐ *strstr.*
- ☐ *strchr.*
- ☐ *strcspn.*

24.

22.11 Q2: What is the output of the following statement? `cout << strspn("Cows like to moo.", "Ceik losw");`

- ☐ *8.*
- ☐ *Nothing.*
- ☐ *10.*
- ☐ *e.*

25.

Section 22.12 Memory Functions of the Pointer-Based String-Handling Library

22.12 Q1: The _____ function allows characters of one part of a string to be copied into another, overlapping part of the same string.

- ☐ *memcpy.*
- ☐ *memcmp.*
- ☐ *memmove.*
- ☐ *memchr.*

26.

22.12 Q2: `memcmp` would return _____ for the call `memcmp("Hi, how are you?", "Hi, how are things?", 6)`.

- ☐ *0.*
- ☐ *A negative number.*
- ☐ *A positive number.*
- ☐ *1.*

27.

22.12 Q3: To change the string "ABCDEFGHI" to "aaaaaFGHI" you would use the _____ function.

- ☐ *memcpy.*
- ☐ *memcmp.*
- ☐ *memset.*
- ☐ *memchr.*

[Clear Answers / Start Over](#)[Submit Answers for Grading](#)

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



Copyright © 1995 - 2010 [Pearson Education](#). All rights reserved.
[Legal Notice](#) | [Privacy Policy](#) | [Permissions](#)