

1. What is **not** one of the phases of the waterfall model of software development?
 - 1) Requirements gathering
 - 2) Implementation
 - 3) Maintenance
 - 4) Using

2. What is the main purpose for the design document we created for project 1?
 - 1) To show the functions and their interfaces
 - 2) To make more work for you to do
 - 3) To clearly indicate all internal coding for the project
 - 4) To give a detailed description of the project

3. What is a data type?
 - 1) A collection of values
 - 2) A piece of data with no subparts
 - 3) A collection of operations and the data that is manipulated
 - 4) The implementation in software

4. What is an abstract data type?
 - 1) The actual implementation for the data type
 - 2) The realization of the data type as a software component
 - 3) The notion of what the data type does
 - 4) The interface for the data type

5. What is a data structure?
 - 1) The actual implementation for the data type
 - 2) The realization of the data type as a software component
 - 3) The notion of what the data type does
 - 4) The interface for the data type

6. How do C++ programmers typically implement data structures?
 - 1) Structures
 - 2) Unions
 - 3) Classes
 - 4) enums

7. Which of the following methods **should always** be implemented by the programmer?
 - 1) Set methods
 - 2) Get methods
 - 3) Default constructor
 - 4) Parameterized constructor

8. Which of the following is a default constructor definition?
 - 1) `MyClass::MyClass(int x, int y);`
 - 2) `void MyClass::MyClass();`
 - 3) `MyClass::MyClass(int = 0, int = 1);`
 - 4) `void MyClass::MyClass(int = 0, int = 1);`

9. What normally goes in a header file?
 - 1) The class' implementation
 - 2) The class' header comment
 - 3) The class' public data
 - 4) The class' declaration

10. What normally goes in a cpp file?
 - 1) The class' implementation
 - 2) The class' header comment
 - 3) The class' public data
 - 4) The class' declaration

11. What is not one of the key words indicating a class's access?
 - 1) `open`
 - 2) `public`
 - 3) `private`
 - 4) `protected`

12. What is the main difference between a class and a struct?
 - 1) The struct is more secure
 - 2) The struct is private by default
 - 3) The struct is public by default
 - 4) The struct is protected by default

13. What should your get functions never return?
 - 1) Any private data
 - 2) A reference to private data
 - 3) `void`
 - 4) A copy of the data

14. How do you initialize a constant piece of data in an object?
 - 1) At the beginning of the implementation put an initializer line
 - 2) In the constructor
 - 3) Through the member initializer list
 - 4) In the constructor's declaration

15. Constant objects can only call which methods?
- 1) Get and print methods
 - 2) Set and get methods
 - 3) Methods with a void return type
 - 4) **Methods that are declared as const**
16. How can a constant object call a constructor; i.e. a method that modifies that data of a constant object?
- 1) It's not really constant
 - 2) **The const-ness doesn't begin until after the constructor**
 - 3) The class' methods are allowed to modify private data
 - 4) It just can!!
17. What can friends to a class do, that non-friends cannot?
- 1) Touch their nose
 - 2) Use member functions
 - 3) **Access private data**
 - 4) Print private data
18. Why might you want a static class member?
- 1) So that all objects of a class can share a piece of data
 - 2) So you won't have to update the same private data in all objects of the class
 - 3) **All of the above**
 - 4) None of the above
19. How do you initialize a static class member?
- 1) **At the beginning of the implementation put an initializer line**
 - 2) In the constructor
 - 3) Through the member initializer list
 - 4) In the constructor's declaration
20. What is a null pointer?
- 1) A pointer that points to a very special place
 - 2) **A pointer that points to nothing**
 - 3) A pointer to a default object that you can't use
 - 4) The same as 0 on every system
21. Why do I prefer the value 0 to the value null?
- 1) I can't remember what to #include
 - 2) **null is system dependent and 0 is not**
 - 3) 0 is less typing
 - 4) You can dereference a pointer pointing to 0, but not null

22. Which of the following declares two pointers to ints?
- 1) `int x, *y;`
 - 2) `int *x, y;`
 - 3) `int xPtr, yPtr;`
 - 4) `int *x, *y;`
23. What does the address of operator do?
- 1) Gives you the memory location of whatever you stick it in front of
 - 2) Allows you to dereference it to get to the data
 - 3) Is not allowed on the left hand side of an assignment, i.e. as an *l-value*
 - 4) Cannot be returned from a function
24. Can you have a pointer to constant data and still increment the data directly?
- 1) Yes
 - 2) No
25. Which of the following correctly indicates a constant pointer to non-constant data?
- 1) `const int * const x;`
 - 2) `const double *x;`
 - 3) `char * const x;`
 - 4) `string x;`
26. What is not a way to refer to the 5 element in array A, assuming `aPtr = A`?
- 1) `*(aPtr+5)`
 - 2) `*(A + 5)`
 - 3) `aPtr[5]`
 - 4) `A += 5`
27. What types of arrays do arrays of pointers allow you to create that you cannot create otherwise?
- 1) Arrays of strings
 - 2) Jagged arrays
 - 3) Arrays of characters
 - 4) Arrays of bools
28. Project 1 was easy?
- 1) True