

All multiple choice questions are equally weighted. You can generally assume that code shown in the questions is intended to be syntactically correct, unless something in the question or one of the answers suggests otherwise.

- 1. The best way to describe cout is that...**
  - A. it is a predefined global variable, an instance of a stream.
  - B. it is a C++ function which replaces printf
  - C. it is new kind of C++ statement, used for output
  - D. it is a reserved word that causes C++ to generate output commands.
  - E. it is an alias for the computer's monitor (screen)
- 2. Consider the following line of C code. Choose the answer which is the best way of achieving the same thing in C++.**

```
#define TURNIPS_PER_TON 4677
```

- A. `const int TURNIPS_PER_TON;`  
`TURNIPS_PER_TON = 4677;`
  - B. `const int TURNIPS_PER_TON = 4677;`
  - C. `const TURNIPS_PER_TON 4677`
  - D. `int TURNIPS_PER_TON;`  
`TURNIPS_PER_TON = 4677;`
  - E. `constant TURNIPS_PER_TON = 4677;`
- 3. Which of the following are examples of abstractions in programming, as discussed in class?**
    - A. the type int
    - B. a struct
    - C. a function
    - D. All of A,B, and C above.
    - E. None of A, B, or C above.
  - 4. What is NOT true about constructors?**
    - A. A constructor can initialize public member data of a class.
    - B. A constructor can initialize private member data of a class.
    - C. Once an object has been constructed, a different (overloaded) constructor could be called on the same object.
    - D. A constructor cannot have a return type, not even void.
    - E. Multiple constructors for a class are legal, and they all have the same name.
  - 5. Which are some reasons to pass objects by reference? For example, to functions like**

```
void foo (MyClass & myvar);
```

- 1. So you can change myvar in foo().**
  - 2. Because it is faster to pass references than the actual object for big objects**
  - 3. So you can call foo without using the & operator on the argument**
- A. 1 only
  - B. 3 only
  - C. 1 and 2
  - D. 2 and 3
  - E. 1, 2, and 3

6. What do we know for sure after the following code:

**program:**

```
int i1, i2;  
char s1[10], s2[10];
```

```
cin >> s1 >> i1 >> s2 >> i2;
```

**input from the keyboard:**

```
Hi  
there  
4 2
```

- A. nothing.
- B. s1 contains "Hi" (that's all we can be sure of).
- C. s1 contains "Hi", i1 is 0, s2 contains "there", i2 is 4.
- D. s1 contains "Hi", s2 contains "there", i2 is 4.
- E. s1 contains "Hi", s2 contains "4", i2 is 2.

7. In a program, the statement:

```
assert (0 <= i && i < 10);
```

- A. Forces its condition to be true by changing the values of variables.
- B. Causes execution to halt if its condition is not true.
- C. Must be commented out before compiling in "release" mode under MSVC.
- D. Will give a compile time error if it is not always true.
- E. Is an example of a practice which usually obscures the code and introduces intolerable run time overhead.

8. Which of the following statements are true about built-in arrays in C++:

**a. Bounds checking is performed, so that error of accessing an element out of the range of the array is detected.**

**b. Arrays are passed by value to functions.**

**c. Functions can be declared to return arrays by value.**

**d. The size of an array can be changed at run time.**

- A. a is true.
- B. b is true.
- C. c is true.
- D. d is true.
- E. None are true.

9. **A "postcondition" of a function**
- A. Is something that the function guarantees to be true when it ends (although perhaps not at every statement along the way)
  - B. is something that the function guarantees to be true at every statement during the function
  - C. is an invariant that must be true during the execution of any loop in the function
  - D. is a condition that must be true at the beginning or the function cannot guarantee correct results
  - E. is a condition that the caller of the function is expected to check after the function is complete.

10. **int twiddle (int \*a, int b, int & c) {**  
    **\*a = \*a + 1;**  
    **b = b + 1;**  
    **c = c + 1;**  
    **return \*a + b + c;**  
**}**

```
void twiddleTest() {  
    int x=0, y=0, z=0;  
    x = twiddle(&x, y, z);  
    cout << x+y+z;  
}
```

**What value is displayed when the last line of twiddleTest executes?**

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 5
11. **Which is NOT an essential feature of a proper recursive function?**
- A. One or more base cases
  - B. One or more recursive cases
  - C. The function calling itself at some point.
  - D. Checking the bases cases before calling any recursive cases
  - E. Not permitted to do any I/O

12. The following function is intended to compute the product of its two arguments. Under what conditions does it work (that is, is logically correct)? Perhaps it works correctly for all values of A and B, or perhaps only for some. Chose the weakest (least restrictive) set of conditions under which the function computes the correct answer.

```
int prod1 (int A, int B) {  
    if (A < B)  
        return prod1 (B, A);  
    if (B < 0)  
        return -(prod1 (A, -B));  
    if (B == 0)  
        return 0;  
    return A + prod1(A, B-1);  
}
```

- A. works for all combinations of integers A and B  
B. works only when (A < B)  
C. works only when (B >= 0)  
D. works only when ((A >=B) && (B >=0))  
E. works only when (B == 0)
13. Assuming that binary search is implemented using a recursive function similar to the one shown in class, about how many recursive calls are made to search an array of 8,000 integers?
- A. 7  
B. 13  
C. 80  
D. 800  
E. 4000

14. `char message[ ] = "aaa bbb ccc ddd";`  
`//one space separates aaa from bbb, etc,`  
`message [5] = '\n' ;`  
`cout << message;`

The result displayed is:

- A. aaa bnb ccc ddd  
B. aaa b b ccc ddd  
C. aaa b  
b ccc ddd  
D. aaa b  
E. aaa

15. `int k;`  
`char ch;`  
`cout << "Please enter year of birth";`

**By mistake, a user might type in a letter instead of a number in response to this question. Which code fragment is on the right track to detecting this situation?**

- A. `cin >> k;`  
`if (cin`  
`//OK`  
`else`  
`//bad -- user didn't type in a number`
- B. `cin >> k;`  
`if (k < 0 || k > 9)`  
`//bad input -- not a number...`  
`...`  
`else`  
`//k is good`
- C. `cin >> ch;`  
`if (ch < '0' || ch > '9')`  
`//bad input --not a number...`  
`else`  
`k = ch; // copy the number to k`
- D. `while (!cin.isgood())`  
`cin >> k;`  
`//leave loop when a number is finally read...`
- E. `if (!cin.eof())`  
`cin >> (int) k; //always cast to an int...`

16. `int spfHelper (int N, int k) {`  
`if (k > N/2)`  
`return N;`  
`if ( (N % k) == 0)`  
`return k;`  
`return spfHelper (N, k+1);`  
`}`  
`int spf (int N) {`  
`return spfHelper (N, 2);`  
`}`

**What value is returned from the call**  
`spf (35)`

- A. 2  
 B. 3  
 C. 4  
 D. 5  
 E. there will be an infinite recursion

17. Suppose  $x$  is the name of a private member variable of a class named  $C$ . What is true?

- A.  $x$  can be referenced within any client program which uses  $C$ .
- B.  $x$  can be used within the implementation of any public method of  $C$ .
- C.  $x$  can be initialized by a constructor of  $C$ .
- A. A only
- B. B only
- C. C only
- D. B and C only
- E. A, B, and C

18. MINDREADING PROBLEM (worth 3 multiple choice questions)

Here is the implementation of a method of a certain class. Assume that this method is correct and uses no global variables or constants of any kind. Write a class declaration that shows as much as you can reasonably deduce about the class, just from this code. Assume the author follows a style guideline similar to the one used in this course.

```
bool
Browser::updateNodeCount(int urlIndex) {
    Browser tempBrowser(5, 475.33);
    bool result;
    tempBrowser.setIndex(&result, urlIndex);
    nodes = 1 + tempBrowser.nodeCount();
    return result;
}
```