

Exam 2

Name _____

Please return question sheets and blue book(s) to your TA

- (1) What is the output of the following
- ```
a = 2;
if (a = 3)
 cout << " a is 3";
else
 cout << " a is not equal 3, a is " << a;
```
- (a) a is 3 (b) a is not equal 3, a is 2  
(c) a is not equal 3, a is 3 (d) a is 2
- (2) In order to use cout and cin, which library must be included?
- (a) <iostream> (b) <cctype>  
(c) <fileio> (d) <stdlib>
- (3) What is the output of the following?
- ```
int a = 2;
int b = 4;
cout << " a divided by b is " << (a/b);
```
- (a) a divided by b is 0.5 (b) a divided by b is 2
(c) a divided by b is 0 (d) compilation error
- (4) Consider the break statement in C++, that is used to exit a control structure. Which one of the following statements is not true?
- (a) The break can be used in *switch* statement to exit the switch construct
(b) The break can be used in *for*, *while*, and *do while* to exit the loop
(c) The break can be used in *if else* statement to exit the *if else* control structure
(d) To terminate a program immediately, we don't use break statement
- (5) What is the output of the statement?
- ```
cout << pow(3,2)/2.0;
```
- (a) 4.0 (b) 9.0  
(c) 4.5 (d) none of the above
- (6) Every C++ program begins execution
- (a) at any function as specified by user (b) at function *main*  
(c) depending on operating system (d) depending on command line parameter
- (7) Which of the following will cause C++ syntax error
- (a) if (c > 5); cout << "c is greater than 5\n";  
(b) if (c < 5); cout << "c is greater than 5\n";  
(c) if (c == 5) cout << "c is equal or greater than 5\n";  
(d) if (c = 5) cout << " c is equal to 5\n";
- (8) Which of the following will not increment x by 1?
- (a) x = x + 1; (b) x++;  
(c) ++x; (d) x += 1;  
(e) None of the above
- (9) Assume integer variables *product*, and x both have value of 5. What are the values after the execution of the following statement?
- ```
product *= x++;
```
- (a) product = 25; and x = 6 (b) product = 25; and x = 5
(c) product = 30; and x = 6 (d) product = 25; and x = 5
(e) none of the above
- (10) Assume integer variables *quotient*, and x both have value of 5. What are the values after the execution of the following statement?
- ```
quotient /= ++x;
```
- (a) quotient = 1; and x = 6 (b) quotient = 1; and x = 5  
(c) quotient = 0.8333; and x = 6 (d) quotient = 0.8333; and x = 5  
(e) none of the above

- (11) What is effect of the following for statement?  
**for (;;) ;**  
(a) compilation error  
(b) it executes once and falls off the loop right away then executes the next statement follows for statement  
(c) the statement never get executed  
(d) it will loop forever
- (12) For C++ function declaration (function prototype), choose a right statement  
(a) It is a must. missing explicit function prototype in the main program file is an error  
(b) It is possible to put all function prototypes in one file and include that file in the main file  
(c) It is unnecessary, compiler is smart enough to find them in all the cases  
(d) None of the above statement is true.
- (13) If a function requires parameters. In the function declaration, select a true statement  
(a) All variables' names and types must be listed.  
(b) Only the names are required to list on the prototype.  
(c) Only the types of the parameters are required if they are pass-by-value.  
(d) Neither types nor names are required.
- (14) For C++ function overloading, choose a true statement  
(a) The maximum number of overloading is 7  
(b) Two overloaded functions can not have exactly same number of parameters.  
(c) Two overloaded functions can have exactly same number of parameters as long as one of their names are different  
(d) Two overloaded functions can have exactly same number of parameters as long as the function signatures are different  
(e) None of the above is statement is true.
- (15) For statement with const keyword in C++, which is true  
(a) The statement must put outside of any function definition  
(b) **int y const = 10;** is a valid C++ statement  
(c) Since one can assign a value to a constant integer after it is declared, so **int const y;** is a valid statement  
(d) All variables declared as const must use all upper case letter otherwise it is a compilation error.  
(e) None of the above is true.
- (16) For function f with a pass-by-reference integer parameter. Which statement will call the function with actual parameter x.  
(a) f(&x);                      (b) f(x&);  
(c) f(x);                        (d) call f(x&);
- (17) For **default** statement in C++ switch control structure  
(a) It must be included in the switch statement  
(b) It must be put at the end after all cases and without break  
(c) It is an optional statement  
(d) None of the above is a true statement.
- (18) What is the value of x: x = ceil(fabs(-8+floor(-5.5))) ;  
(a) -13                          (b) 13  
(c) 14                            (d) -14
- (19) For input and output file processing, which one of the following is not true  
a) The <fstream> shall be included  
b) For a file to open without discarding the current data and put the new data to the end, one can use ios::app as one of the parameter for method open.  
c) It would be a syntax error if a file is opened but the method close never showed in your program  
d) To call function close after you open a file is a good practice but not a requirement for writing a valid C++ program.

- (20) For a scope rule and life span of variables, which is not true
- a) A static variable of a function only initialize once
  - b) A global constant shall be declared inside the main function
  - c) Variables declared inside a function have are called local variable and cannot access outside the function
  - d) A pass-by-value parameter of a function is treated as a local variable
- (21) When a compiler encounters a function parameter for a single-subscripted array of the form `int a[]`, it converts the parameter to:
- a) `int a`
  - b) `int &a`
  - c) `int *a`
  - d) No conversion is necessary.
- (22) Which of the following is *not* a correct way to initialize a built-in array?
- a) `int n[ 5 ] = { 0, 7, 0, 3, 8, 2 };`
  - b) `int n[] = { 0, 7, 0, 3, 8, 2 };`
  - c) `int n[ 5 ] = { 7 };`
  - d) `int n[ 5 ] = { 9, 1, 9 };`
- (23) To *prevent* modification of a built-in array's values when you pass the built-in array to a function:
- a) The built-in array must be declared static in the function.
  - b) The built-in array parameter can be preceded by the `const` qualifier.
  - c) A copy of the built-in array must be made inside the function.
  - d) The built-in array must be passed by reference.
- (24) Which of the following is *false*?
- a) The last element of an array has position number one less than the array size.
  - b) The position number contained within square brackets is called a subscript.
  - c) A subscript cannot be an expression.
  - d) All of the above.
- (25) Which statement would be used to declare a 12-element integer array `c`?
- a) `array c< 12 >;`
  - b) `array c< int, 12 >;`
  - c) `array< 12 > c;`
  - d) `array< int, 12 > c;`
- (26) Which of the following is *not* true of class template `vector`?
- a) The size of a vector can be changed after it is declared.
  - b) A vector can be assigned to another vector by using the assignment operator.
  - c) A vector object can be initialized with a copy of another vector by invoking the copy constructor.
  - d) A vector can store only data of type `int`.
- (27) When using exception handling, place any code that might *throw* an exception in a \_\_\_\_\_.
- a) `catch` block
  - b) `try` statement.
  - c) `throw` block.
  - d) `what` statement.
- (28) What does the following statement declare?
- ```
int *countPtr, count;
```
- a) Two `int` variables.
 - b) One pointer to an `int` and one `int` variable.
 - c) Two pointers to `ints`.
 - d) The declaration is invalid.
- (29) All of the following *can* cause a fatal execution-time error *except*:
- a) Dereferencing a pointer that has not been assigned to point to a specific address.
 - b) Dereferencing a pointer that has not been initialized properly.
 - c) Dereferencing a null pointer.
 - d) Dereferencing a variable that is not a pointer.

- (30) A function that modifies an array by using pointer arithmetic such as `++ptr` to process every value of the array should have a parameter that is:
- a) A nonconstant pointer to nonconstant data.
 - b) A nonconstant pointer to constant data.
 - c) A constant pointer to nonconstant data.
 - d) A constant pointer to constant data.
- (31) A function that prints a string by using pointer arithmetic such as `++ptr` to output each character should have a parameter that is:
- a) A nonconstant pointer to nonconstant data.
 - b) A nonconstant pointer to constant data.
 - c) A constant pointer to nonconstant data.
 - d) A constant pointer to constant data.
- (32) `sizeof`:
- a) Is a binary operator.
 - b) Returns the total number of elements in an array.
 - c) Usually returns a double.
 - d) Returns the total number of bytes in a variable.
- (33) Which of the following *can* have a pointer as an operand?
- a) `++`
 - b) `*=`
 - c) `%`
 - d) `/`
- (34) 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 + 5` point to?
- a) 2003
 - b) 2006
 - c) 2012
 - d) 2020
- (35) Assuming that `t` is an array and `tPtr` is a pointer to that array, which expression refers to the address of 4th element of the array?
- a) `*(tPtr + 3)`
 - b) `tPtr[3]`
 - c) `&t[3]`
 - d) `*(t + 3)`
- (36) Which of the following preprocessor directives does *not* constitute part of the preprocessor wrapper?
- a) `#define`
 - b) `#endif`
 - c) `#ifndef`
 - d) `#include`
- (37) If the line:
`friend class A;`
appears in class B, and the line:
`friend class B;`
appears in class C, then:
- a) Class A is a friend of class C.
 - b) Class A can access private variables of class B.
 - c) Class C can call class A's private member functions.
 - d) Class B can access class A's private variables.
- (38) Which statement about operator overloading is *false*?
- a) Operator overloading is the process of enabling C++'s operators to work with class objects.
 - b) C++ overloads the addition operator (+) and the subtraction operator (-) to perform differently, depending on their context in integer, floating-point and pointer arithmetic with data of fundamental types.
 - c) You can overload all C++ operators to be used with class objects.
 - d) When you overload operators to be used with class objects, the compiler generates the appropriate code based on the types of the operands.

- (39) The delete [] operator:
- Can terminate the program.
 - Must be told which destructor to call when destroying an object.
 - Can delete an entire array of objects declared using new.
 - Is called *implicitly* at the end of a program.
- (40) Because the postfix increment operator returns objects by value and the prefix increment operator returns objects by reference:
- Prefix increment has slightly more overhead than postfix increment.
 - The postfix increment operator returns the actual incremented object with its new value.
 - Objects returned by postfix increment cannot be used in larger expressions.
 - The postfix increment operator typically returns a temporary object that contains the original value of the object before the increment occurred.

Program coding (each 12 points) partial credits will be given if not completed

1. Write a program segment which will ask user to enter a number, n , between 1 and 10, your program will output from 1 to n , the number, its square, its cube in a formatted fashion. For example, if user enter 4, then your program segment will output something like the following.

1	1	1
2	4	8
3	9	27
4	16	64

2. Write a program segment that will output 10 random numbers between 1 and 6. (just write program segment not the whole program). You may use the traditional rand() function in this problem. Or you may use more recent C++11 default_random_engine class and uniform_int_distribution template class to generate random numbers. The rand() apparently is simpler, but you can make your own choice.

3. Write a C++ function, reverse, which will take two parameters: one is a C build-in integer array and the other is the size of this integer array. Your function will reverse the content of the array. For example. If you have an array A which contains [1,3,5,7,9], then after you calling reverse(A, 5), the content of A will become [9,7,5,3,1]. Implement just this function reverse.

4. Write a program that will compute the standard deviation of an array of type double. Print out the numbers and the standard deviation. Note standard deviation of numbers x_1, x_2, \dots, x_N is defined as the following

$$S = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}}$$

Where \bar{x} is the average of these N numbers.

5. Define a class Rational using C++ as shown in the class. Overload operators, +, -, *, /, =, ==, and != as member functions. So that we if x , y , and z are rational numbers then we can write statement such as $x = y + z$, $x = y$, $x = y * z$ etc. we can also compare two numbers such as $x == y$, $x != y$ etc. Overload << and >> as non member functions. The << operator will allow you to output rational numbers (allow cascading). You can decide your input format when you overload >> operator.