

Sample Exam Week 01

CSE 232 (Introduction to Programming II)

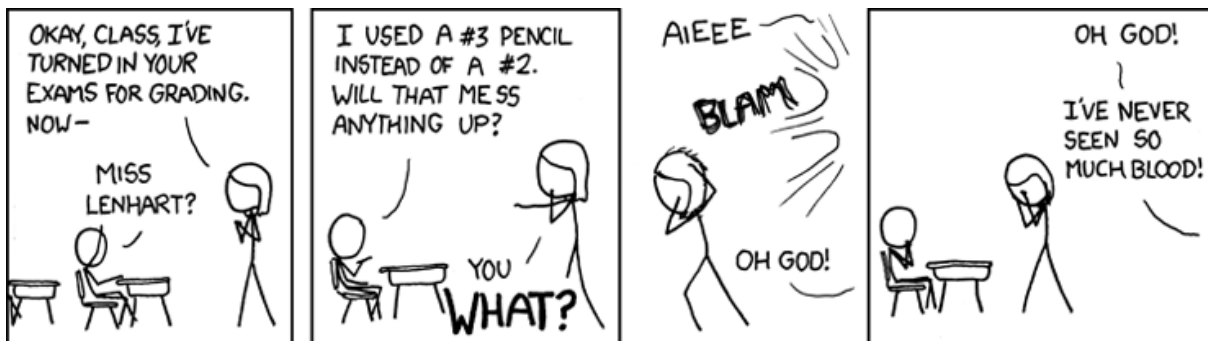
VERSION A

Full Name:

Student Number:

Instructions:

- DO NOT START/OPEN THE EXAM UNTIL TOLD TO DO SO.
- You may however write and bubble in your name, student number and exam **VERSION/FORM NUMBER** (with a #2 pencil) on the front of the printed exam and bubble sheet prior to the exam start. This exam is Version A. Your section doesn't matter and can be ignored.
- Present your MSU ID (or other photo ID) when returning your bubble sheet and printed exam.
- Only choose one option for each question. Please mark the chosen option in both this printed exam and the bubble sheet.
- Assume any needed `#includes` and `using std::...;` namespace declarations are performed for the code samples.
- Every question is worth the same amount of points. There are 55 questions, but you only need 50 questions correct for a perfect score.
- No electronics are allowed to be used or worn during the exam. This means smart-watches, phones and headphones need to be placed away in your bag.
- The exam is open note, meaning that any paper material (notes, slides, prior exams, assignments, books, etc.) are all allowed. Please place all such material on your desk prior to the start of the exam, (so you won't need to rummage in your bag during the exam).
- If you have any questions during the exam or finish the exam early, please raise your hand and a proctor will attend you.



<http://xkcd.com/499/>

1. When should you use a postfix increment instead of a prefix increment?
 - (a) When incrementing a value will result in side effects beyond the expression itself
 - (b) When the compiler warns of ambiguous or undefined behavior
 - (c) When you need the value returned by the expression to be the pre-increment value
 - (d) When the type of the value being incremented is an integer or character, or if the value should be interpreted as such
 - (e) When clarity is more important than performance

2. Which of the following is a bitwise operator?
 - (a) `operator||`
 - (b) `operator*`
 - (c) `operator=`
 - (d) `operator++`
 - (e) `operator&`

3. Which of the following operations can you do with a constant variable?
 - (a) Initialize It
 - (b) Assign To It
 - (c) Print It
 - (d) Change It
 - (e) You can do two of the above
 - (f) You can do three of the above
 - (g) You can do all of the above

4. An identifier that is not declared inside any other construct has what scope?
 - (a) total
 - (b) undefined
 - (c) reserved
 - (d) file
 - (e) global
 - (f) program
 - (g) local

5. Which of the following is NOT a benefit to using curly braces for initialization?
 - (a) Allows const variables to be initialized
 - (b) Can't be confused for a function declaration
 - (c) Avoids the "Most Vexing Parse"
 - (d) Avoids narrowing conversions

6. Why should undefined behavior be avoided?
 - (a) Because it means that your program must be run with only the most modern of compilers
 - (b) Because it means that your program can't compile
 - (c) Because it means that your program will be difficult to write and even more difficult to read
 - (d) Because it means that your program's could do anything
 - (e) Because it means that your program can only run on specific hardware

7. Why does C++ allow for programs with undefined behavior?
 - (a) Because it allows for compiled programs to run more efficiently
 - (b) Because detecting undefined behavior is difficult and/or expensive
 - (c) Because it expects that software developers will be able to write code that doesn't produce undefined behavior
 - (d) All of the above are true

8. What is the value of `x`?


```
bool x{0123 == 83};
```

 - (a) `false`
 - (b) `bool`
 - (c) `true`
 - (d) `83`
 - (e) Impossible to determine with the information given

9. If you wanted a type that was an integer of 32 bits in size, which type should you use?
- double
 - int
 - char
 - unsigned
 - None of the above
10. Which of the following is equivalent to this statement? Note, the type of `other` isn't specified.
- ```
double const d{other};
```
- const double d{other};
  - double const d = other;
  - const double d = other;
  - All of the above are equivalent
11. What is the value of `x`?
- ```
int y = 200;
int x = (45 <= y <= 100);
```
- 0
 - 1
 - 45
 - 100
 - 200
12. What is the result of the expression `10 / 3`?
- 3
 - 4
 - 3.333333334
 - 1
 - Depends on the type of 10 and 3
 - Depends on the compiler environment
13. Which of the following expressions generate type errors?
- `0xAA + 5`
 - `054 + '2'`
 - `5.6 + 3`
 - All of (a-c) generate type errors.
 - None of (a-c) generate type errors.
14. Why (generally) should you avoid starting integer with leading zeros (i.e. 0099)?
- Leading zeroes cause confusion because they are mistaken for the letter '0'.
 - Leading zeroes are redundant.
 - Leading zeroes are fine, they aren't a problem.
 - Leading zeroes indicate octal literals.
 - Leading zeroes reduce readability.
15. What is printed by the following code?
- ```
char x = 'a' + 2; std::cout <<
x;
```
- 'a'2
  - c
  - a2
  - None of the above (because it causes a type conversion error).
16. Assessing the value of an uninitialized `int` causes what to happen?
- An exception to be thrown at run-time.
  - Undefined behavior
  - A compiler error
  - The value to become 0
17. Which of the following is equivalent to the C++ expression:
- ```
(0 <= x < 7)
```
- `(0 <= x) % (x < 7)`
 - `(0 <= x) || (x < 7)`
 - `(0 <= x) && (x < 7)`
 - `(0 <= x) * (x < 7)`
 - `(0 <= x) < 7`
 - None of the above.

18. What does the following code output?
- ```
char c = 'a';
char d, e;
d = e = c;
c = 'b';
cout << c << d << e;
```
- (a) cde
  - (b) The code will not compile.
  - (c) aab
  - (d) bab
  - (e) bbb
  - (f) baa
  - (g) None of the above.
19. What is the path of the root directory?
- (a) ..
  - (b) .
  - (c) /
  - (d) ~
  - (e) /root
  - (f) /home
  - (g) None of the above are correct.
20. What is this program's output?
- ```
int x = 3, y = 2;
y = x++ + y;
y *= 2;
cout << y / x;
```
- (a) 2
 - (b) 2.5
 - (c) 3
 - (d) Undefined Behaviour
 - (e) None of the above are correct.
21. When declaring an int, what is its initial value?
- (a) Its address
 - (b) undefined
 - (c) -1
 - (d) Depends on if the int is declared in a loop or not.
 - (e) 0
 - (f) None of the above
22. The `clang-format` tool adjusts what property of the files it is given?
- (a) It does none of the above.
 - (b) It separates function definitions from function declarations to make separate compilation easier to accomplish.
 - (c) It adjusts the whitespace (indentation and spacing).
 - (d) It renames variables to match the style guide.
 - (e) It checks for C++ standards violations (like comparing signed and unsigned ints).
23. What is the value of y after these lines are run in C++?
- ```
int x = 3;
double y = x / 4;
```
- (a) 0
  - (b) 0.75
  - (c) 1
  - (d) 3
  - (e) 4
  - (f) Undefined
24. Assuming that x, y, and z are all defined as integers, which of the following is true in C++:
- (a) `[x, y]` will convert x and y into coordinates
  - (b) `!!y` will turn y into a 0 or 1 (false or true) through double negation.
  - (c) `x == z` will tell you if x is either equal to or greater-than z
  - (d) `x < y < z` will tell you if x, y, and z are in order from smallest to largest
  - (e) More than one of the answers above is true

25. What is the value of `b` after the following code executes?
- ```
int a = 2;
int b = 4 * a;
std::cout << (b += 2);
```
- (a) 2
 - (b) 4
 - (c) 6
 - (d) 10
 - (e) 12
 - (f) 14
 - (g) Nothing: it is illegal C++ code
26. What is the output of the following code?
- ```
int x = 1, y = 2;
std::cout << (++x * y++) <<
std::endl;
```
- (a) 2
  - (b) 3
  - (c) 6
  - (d) 1
  - (e) 4
27. What is the value of `y`?
- ```
double y = 13 / 2;
```
- (a) 6.5
 - (b) 6.0
 - (c) The code won't compile.
28. What symbol/identifier denotes the home directory?
- (a) /
 - (b) ..
 - (c) -
 - (d) ~
 - (e) .
29. What is the value of `x` in the following program?
- ```
auto x = 'b' - 'a';
```
- (a) 1
  - (b) It is the ASCII value for the character 'c'
  - (c) -1
  - (d) 'a'
  - (e) 'b'
30. Which of the following are legal comments in C++
- (a) `//// This is a legal comment`
  - (b) All of the above
  - (c) `//--- This is a legal comment`  
`---//`
  - (d) `// This is a legal comment`
  - (e) `/* This is a legal comment */`
31. Why does the following code generate a compiler error?
- ```
char c = '0';
cin >> c;
cout << ++c;
```
- (a) No error is caused, the code should compile.
 - (b) The extraction must fail because `std::skipws` wasn't used, so the variable `c` can only be assigned a whitespace character.
 - (c) Char variables must be initialized with characters, not digits.
 - (d) You can't use the extraction operator on an initialized variable.
 - (e) The pre-increment operator can only be used on integer types (i.e. not with a char).
32. How large is an `int`?
- (a) 1 byte
 - (b) Depends on the system
 - (c) 8 bytes
 - (d) 4 bytes
 - (e) 2 bytes

33. What is NOT included when initializing a variable?
- The variable's name
 - The variable's type
 - None of the above
 - The variable's value
34. Which of the following types is largest?
- Multiple types are tied for largest.
 - `char`
 - `int`
 - Depends on compiler/OS.
 - `short`
 - `long`
 - `bool`
 - `long long`
35. `cd ..` changes your current working directory to what directory?
- The directory that was previously used
 - The parent directory
 - The directory that contains the currently executing program
 - The directory named `".."`
 - The home directory
36. On my system, when I call `sizeof(bool)`; I get the value 1. What does this mean?
- It means that a `bool` takes up one bit.
 - It means that a `bool` true value is equivalent to an integer 1.
 - It means that a `bool` can hold at most one value.
 - It means that a `bool` takes up one byte.
 - It means that a `bool` takes up one word (on a 64-bit machine this is 64 bits).
 - It means that a `bool`'s size is the same as the size of a function.
 - It means that a `bool`'s type determines if it is true or false.
37. For which values of `int x` will the following expression be true?
`-2 < x <= 2`
- 2, -1, 0, 1
 - None exist
 - 1, 0, 1, 2
 - 2, -1, 0, 1, 2
 - 1, 0, 1
 - All possible values of `x`
38. Which of the following is NOT a binary operator?
- `/`
 - `=`
 - `>=`
 - `+=`
 - `++`
 - `<<`
 - `&&`
39. What is wrong with this function invocation?
`std::cout << Func(int x, &y);`
- The code will result in undefined behavior due to the lack of assignment
 - `Func` is an invalid name due to the presence of an uppercase letter
 - Functions can't be invoked in larger expressions
 - The `std::endl` was omitted
 - The namespace for the function isn't specified
 - The ampersand (`&`) doesn't belong in a function call
 - Arguments shouldn't have a type declaration
40. Accessing the value of an uninitialized `char` variable will result in what?
- A segmentation fault
 - A null pointer
 - A random character from the ASCII chart
 - A compile-time error
 - Undefined behavior
 - The space character (`' '`)
 - A null character

41. What does the `std::endl` object do when passed as the second operand to the put to (`<<`) operator?
- It resets the stream
 - It indicates that the stream should be prepared for new characters
 - It doesn't do anything, it is instead used to indicate comments
 - It causes the stream to separate words according to whitespace
 - It causes a newline character to be written to the stream
 - It can't be used with an insertion operator
 - It indicates the End-Of-File
42. What arguments is this function being called with?
- ```
int x = 4; int y = 17;
Func(++x, y--);
```
- 5 and 17
  - 4 and 16
  - 5 and 16
  - 4 and 17
43. How do you make a variable that can have multiple types?
- By declaring its type as `auto`
  - By using `static_cast`
  - By using `decltype`
  - By using `typedef`
  - It is impossible
44. What is the type of `x`?
- ```
auto x = My_Function("abcd");
```
- `void`
 - `int`
 - `char`
 - `std::string`
 - Impossible to determine with the information given
45. What is the value of `y` after these lines are run in C++?
- ```
int x = 3;
double y = x / 4;
```
- 0
  - 0.75
  - 1
  - 3
  - 4
  - Undefined
46. If a function is said to be overloaded, that means that there are two functions that share what property?
- The same body (using templates)
  - The same return type
  - The same parameters
  - The same name
  - The same libraries
  - The same header file
  - The same arguments
47. Which of the following types are NOT fundamental types?
- `double`
  - `bool`
  - `char`
  - `unsigned`
  - `int`
  - None of the above
48. What distinguishes the prefix increment operator from the postfix increment operator?
- The prefix increment operator is part of the C++ standard.
  - The prefix increment returns the value after the increment occurs.
  - The prefix increment operator increments ints and double variables.
  - The prefix increment operator can be used in larger expressions.
  - The prefix increment operator only increments non-zero values.
  - None of the above

49. Curly braces in initialization (e.g. `int x {7};`), have what advantage over `'='` initialization (e.g. `int x = 7;`)?
- Implicit narrowing conversions are disallowed.
  - Constant initialization can occur.
  - The variable has local scope.
  - They are able to be used on user-defined types.
  - Compiler checks that initialization is zero initialized.
  - None of the above
50. When should you use type inference?
- When the definition is in a large scope.
  - When the type of the initializer isn't obvious.
  - The variable has local scope.
  - When you want to be explicit about a variable's range or precision.
  - When you want to make the type clearly visible to readers.
  - None of the above
51. The `const` in an initialization implies what characteristic of the variable is constant?
- Its representation
  - Its meaning
  - Its location
  - Its value
  - Its type
  - None of the above
52. Which of the following declarations follow the "East const Style"?
- `const Thing t;`
  - `Thing_const t;`
  - `Thing t_const;`
  - `Thing t const;`
  - `Thing const t;`
  - None of the above
53. The `"-i"` flag when given to `clang-format` changes its behavior in what way?
- It does nothing in most cases, but causes the formatter to raise an error if the code is invalid C++.
  - It allows for an interactive formatting where the edits can be reviewed by the user.
  - It replaces the file with the formatted version.
  - It allows for internal representations to be used, which results in more efficient code to be generated.
  - It ensures that the formatted output is case-insensitive.
  - None of the above
54. What is the result of the following expression?  
`7 % 5`
- 0
  - 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - 7
55. If your current working directory is named "Lectures", which is located in a folder named "CSE232". How do you change the current working directory to be "CSE232"?
- `cd CSE232`
  - `cd ~`
  - `cd ..`
  - `cd`
  - `cd Lectures\CSE232`
  - `cd \`
  - None of the above



56. The `clang-format` tool adjusts what property of the files it is given?

- (a) It renames variables to match the style guide.
- (b) It adjusts the whitespace (indentation and spacing).
- (c) It does none of the above.
- (d) It separates function definitions from function declarations to make separate compilation easier to accomplish.
- (e) It checks for C++ standards violations (like comparing signed and unsigned ints).

