

Sample Exam Week 01

CSE 232 (Introduction to Programming II)

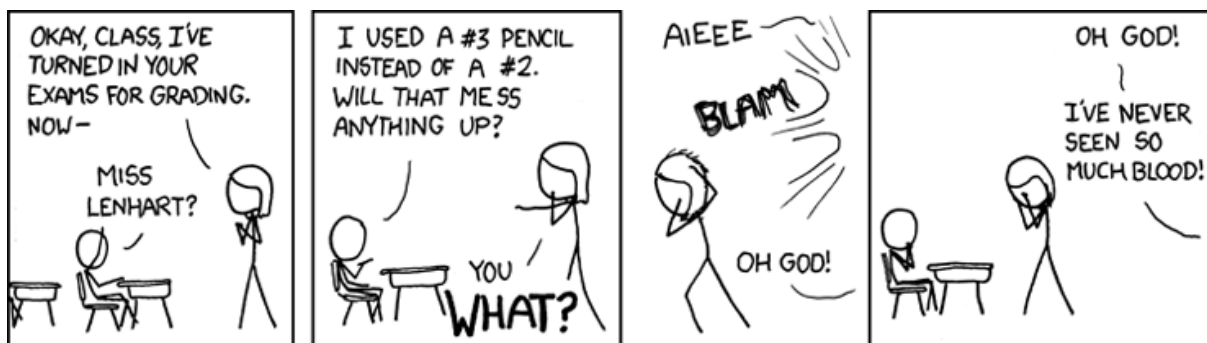
VERSION A

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. When should `auto` be used in an initialization?
 - (a) When writing out the name of the type would be laborious
 - (b) When the type of a variable isn't known
 - (c) When the compiler indicates that it is necessary
 - (d) All of the above are true

9. What is the value of `x`?
`bool x{0123 == 83};`
- `true`
 - `false`
 - `bool`
 - `83`
 - Impossible to determine with the information given
10. If you wanted a type that was an integer of 32 bits in size, which type should you use?
- `unsigned`
 - `char`
 - `double`
 - `int`
 - None of the above
11. Which of the following is equivalent to this statement?
`double const d{other};`
- `double const d = other;`
 - `const double d = other;`
 - `const double d{other};`
 - All of the above are equivalent
12. What is the value of `x`?
`int y = 200;`
`int x = (45 <= y <= 100);`
- 200
 - 100
 - 45
 - 1
 - 0
13. What is the result of the expression `10 / 3`?
- 3
 - Depends on the type of 10 and 3
 - 3.333333334
 - Depends on the compiler environment
 - 4
 - 1
14. 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.
15. Why (generally) should you avoid starting integer with leading zeros (i.e. 0099)?
- Leading zeroes reduce readability.
 - Leading zeroes are redundant.
 - Leading zeroes are fine, they aren't a problem.
 - Leading zeroes indicate octal literals.
 - Leading zeroes cause confusion because they are mistaken for the letter 'O'.
16. What is printed by the following code?
`std::cout << 'a' + 2;`
- `c`
 - `'a'2`
 - `a2`
 - None of the above (because it causes a type conversion error).
17. Assessing the value of an uninitialized `int` causes what to happen?
- An exception to be thrown at run-time.
 - The value to become 0
 - A compiler error
 - Undefined behavior
18. 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.

19. What does the following code output?

```
char c = 'a';  
char d, e;  
d = e = c;  
c = 'b';  
cout << c << d << e;
```

- (a) cde
- (b) baa
- (c) The code will not compile.
- (d) bab
- (e) bbb
- (f) aab
- (g) None of the above.

20. What is the path of the root directory?

- (a) ..
- (b) /root
- (c) .
- (d) /home
- (e) /
- (f) ~
- (g) None of the above are correct.

21. 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) Undefined Behaviour
- (d) 3
- (e) None of the above are correct.

22. When declaring an int, what is its initial value?

- (a) Its address
- (b) undefined
- (c) -1
- (d) 0
- (e) Depends on if the int is declared in a loop or not.
- (f) None of the above

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

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

24. 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

25. Assuming that x, y, and z are all defined as integers, which of the following is true in C++:

- (a) `x == z` will tell you if x is either equal to or greater-than z
- (b) `x < y < z` will tell you if x, y, and z are in order from smallest to largest
- (c) `[x, y]` will convert x and y into coordinates
- (d) `!!y` will turn y into a 0 or 1 (false or true) through double negation.
- (e) More than one of the answers above is true

26. 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

27. What is the output of the following code?

```
int x = 1, y = 2;
std::cout << (++x * y++) <<
std::endl;
```

- (a) 4
- (b) 3
- (c) 2
- (d) 1
- (e) 6

28. What is the value of **y**?

```
double y = 13 / 2;
```

- (a) The code won't compile.
- (b) 6.5
- (c) 6.0

29. What symbol/identifier denotes the home directory (the only directory with no parent)?

- (a) ..
- (b) ~
- (c) -
- (d) /
- (e) .

30. What is the value of **x** in the following program?

```
auto x = 'b' - 'a';
```

- (a) It is the ASCII value for the character 'c'
- (b) 'a'
- (c) -1
- (d) 1
- (e) 'b'

31. Which of the following are legal comments in C++

- (a) `// This is a legal comment`
- (b) `/* This is a legal comment */`
- (c) `/-- This is a legal comment`
`---//`
- (d) `//// This is a legal comment`
- (e) All of the above

32. Why does the following code generate a compiler error?

```
char c = '0';
cin >> c;
cout << ++c;
```

- (a) Char variables must be initialized with characters, not digits.
- (b) You can't use the extraction operator on an initialized variable.
- (c) The pre-increment operator can only be used on integer types (i.e. not with a char).
- (d) No error is caused, the code should compile.
- (e) The extraction must fail because `std::skipws` wasn't used, so the variable `c` can only be assigned a whitespace character.

33. How large is an `int`?

- (a) 1 byte
- (b) Depends on the system
- (c) 8 bytes
- (d) 2 bytes
- (e) 4 bytes

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

41. Accessing the value of an uninitialized `char` variable will result in what?
- (a) A segmentation fault
  - (b) Undefined behavior
  - (c) A null character
  - (d) The space character ( ' ' )
  - (e) A compile-time error
  - (f) A random character from the ASCII chart
  - (g) A null pointer
42. What does the `std::endl` object do when passed as the second operand to the insertion operator?
- (a) It indicates the End-Of-File
  - (b) It resets the stream
  - (c) It causes the stream to separate words according to whitespace
  - (d) It doesn't do anything, it is instead used to indicate comments
  - (e) It causes a newline character to be written to the stream
  - (f) It indicates that the stream should be prepared for new characters
  - (g) It can't be used with an insertion operator
43. What arguments is this function being called with?
- ```
int x = 4; int y = 17;
Func(++x, y--);
```
- (a) 4 and 17
 - (b) 5 and 16
 - (c) 5 and 17
 - (d) 4 and 16
44. How do you make a variable that can have multiple types?
- (a) By using `typedef`
 - (b) By declaring its type as `auto`
 - (c) By using `static_cast`
 - (d) By using `decltype`
 - (e) It is impossible
45. What is the type of `x`?
- ```
auto x = My_Function("abcd");
```
- (a) `char`
  - (b) `int`
  - (c) `std::string`
  - (d) `void`
  - (e) Impossible to determine with the information given
46. 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
47. If a function is said to be overloaded, that means that there are two functions that share what property?
- (a) The same header file
 - (b) The same return type
 - (c) The same arguments
 - (d) The same name
 - (e) The same parameters
 - (f) The same body (using templates)
 - (g) The same libraries

