Week 01 Sample Exam

CSE 232 (Introduction to Programming II)

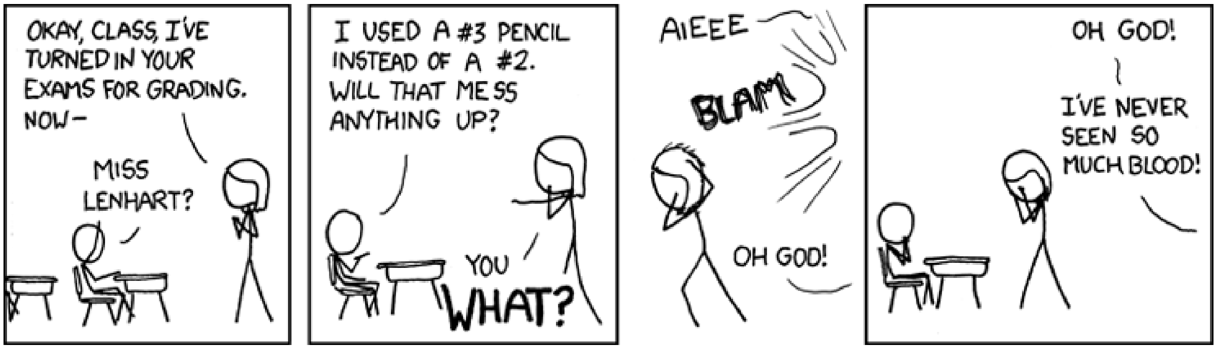
VERSION A

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 LETTER** (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 `#include`s 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). Please be sure to bring the required textbook!
- If you have any questions during the exam or when you finish the exam, please raise your hand and a proctor will attend you.


OKAY, CLASS, I'VE TURNED IN YOUR EXAMS FOR GRADING. NOW—
MISS LENHART?
I USED A #3 PENCIL INSTEAD OF A #2. WILL THAT MESS ANYTHING UP?
YOU WHAT?
AIEEE
BLAM
OH GOD!
OH GOD!
I'VE NEVER SEEN SO MUCH BLOOD!

http://xkcd.com/499/ Date Accessed: October 16, 2024

1. For the following for loop, what would happen if the postfix increment was changed to a prefix increment?

```
for (int i{0}; i < x; i++) ...
```

   (a) The loop would iterate one more time
   (b) The loop would iterate one fewer time
   (c) The final value of `i` would be one smaller
   (d) The value of `i` would be one larger in the body of the for loop
   (e) The value of `i` would be one smaller in the body of the for loop
   (f) The final value of `i` would be one larger
   (g) None of the above are true

2. What is the path of the parent directory?

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

3. What is the proper syntax to import a library in C++? Assume the library is called 'cse232'.

   (a) `#include <cse232>`
   (b) `using cse232;`
   (c) `include <cse232>`
   (d) `#include cse232`
   (e) `include cse232`
   (f) `using "cse232";`

4.
```
std::cout << a << '\n';
a++;
```
The above code fragment causes this error:
```
error: increment of read-only variable
'a'
10 | a++;
```
What is a possible variable declaration for `a` that produces the above results?

   (a) `int const a = 5;`
   (b) `int * a = &c;`
   (c) `int a = 5;`
   (d) `auto a = 5;`
   (e) `int a;`

5. This code causes an compiler error. Select the reason why.

```
int sum = 0;
for(int x = 0, x < 7, x++){
  sum += x;
}
std::cout << sum << '\n';
```

   (a) `++x` should have been used over `x++`
   (b) The `+=` operator cannot be used with variables on the right hand side, only with literals
   (c) `std::cout` can't be used with `operator<<`, only `operator>>`
   (d) For loop parameters should be separated with semicolons, not commas
   (e) `x` should have been declared outside the for loop
   (f) The `sum` variable can't be used inside the for loop

6. When should you use a postfix increment instead of a prefix increment?

   (a) When clarity is more important than performance
   (b) When you need the value returned by the expression to be the pre-increment value
   (c) When the compiler warns of ambiguous or undefined behavior
   (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 incrementing a value will result in side effects beyond the expression itself

7. Which of the following is a bitwise operator?

   (a) `operator||`
   (b) `operator*`
   (c) `operator=`
   (d) `operator&`
   (e) `operator++`

8. 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

9. An identifier that is not declared inside any other construct has what scope?

   (a) file
   (b) local
   (c) global
   (d) total
   (e) program
   (f) reserved
   (g) undefined

10. Which of the following is NOT a benefit to using curly braces for initialization?

    (a) Avoids the ”Most Vexing Parse”
    (b) Allows const variables to be initialized
    (c) Avoids narrowing conversions
    (d) Can’t be confused for a function declaration

11. Why should undefined behavior be avoided?

    (a) Because it means that your program can’t compile
    (b) Because it means that your program’s could do anything
    (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 can only run on specific hardware
    (e) Because it means that your program must be run with only the most modern of compilers

12. Why does C++ allow for programs with undefined behavior?

    (a) Because it allows for compiled programs to run more efficiently
    (b) Because it expects that software developers will be able to write code that doesn’t produce undefined behavior
    (c) Because detecting undefined behavior is difficult and/or expensive
    (d) All of the above are true

13. What is the value of `x`?
    `bool x{0123 == 123};`

    (a) `true`
    (b) `123`
    (c) `false`
    (d) `bool`
    (e) Impossible to determine with the information given

14. If you wanted a type that was an integer of 32 bits in size, which type should you use?

    (a) `double`
    (b) `int`
    (c) `char`
    (d) `unsigned`
    (e) None of the above

15. Which of the following is equivalent to this statement? Note, the type of `other` isn't specified.
    `double const d{other};`

    (a) `const double d{other};`
    (b) `double const d = other;`
    (c) `const double d = other;`
    (d) All of the above are equivalent

16. What is the value of `x`?
    `int y = 200;`
    `int x = (45 <= y <= 100);`

    (a) `200`
    (b) `100`
    (c) `45`
    (d) `1`
    (e) `0`

17. What is the result of the expression `10 / 3`?

    (a) Depends on the type of `10` and `3`
    (b) 1
    (c) Depends on the compiler environment
    (d) 3.333333334
    (e) 4
    (f) 3

18. Which of the following expressions generate type errors?

    (a) `0xAA + 5`
    (b) `054 + '2'`
    (c) `5.6 + 3`
    (d) All of (a-c) generate type errors.
    (e) None of (a-c) generate type errors.

19. Why (generally) should you avoid starting integer with leading zeros (i.e. 0099)?

    (a) Leading zeroes are redundant.
    (b) Leading zeroes indicate octal literals.
    (c) Leading zeroes cause confusion because they are mistaken for the letter `'O'`.
    (d) Leading zeroes are fine, they aren't a problem.
    (e) Leading zeroes reduce readability.

20. What is printed by the following code?
    `char x = 'a' + 2;`
    `std::cout << x;`

    (a) `c`
    (b) `a2`
    (c) `'a'2`
    (d) None of the above (because it causes a type conversion error).

21. Assessing the value of an uninitialized `int` causes what to happen?

    (a) Undefined behavior
    (b) An exception to be thrown at run-time.
    (c) The value to become `0`
    (d) A compiler error

22. Which of the following is equivalent to the C++ expression:
    `(0 <= x < 7)`

    (a) `(0 <= x) || (x < 7)`
    (b) `(0 <= x) % (x < 7)`
    (c) `(0 <= x) * (x < 7)`
    (d) `(0 <= x) && (x < 7)`
    (e) `(0 <= x) < 7`
    (f) None of the above.

23. What does the following code output?

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

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

24. What is the path of the root directory?

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

25. What is this program’s output?

```
int x = 3, y = 2;
y = x++ + y;
y *= 2;
cout << y / x;
```

- (a) 2.5
- (b) 2
- (c) Undefined Behaviour
- (d) 3
- (e) None of the above are correct.

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

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

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

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

28. 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

29. 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

30. 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

31. What is the output of the following code?

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

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

32. What is the value of y?

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

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

33. What symbol/identifier denotes the home directory?

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

34. What is the value of x in the following program?

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

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

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

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

36. 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) The pre-increment operator can only be used on integer types (i.e. not with a char).
(d) Char variables must be initialized with characters, not digits.
(e) You can't use the extraction operator on an initialized variable.

37. How large is an int?

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

38. What is NOT included when initializing a variable?

(a) The variable's type
(b) The variable's value
(c) None of the above
(d) The variable's name

39. Which of the following types is largest?

    (a) `char`
    (b) `int`
    (c) `long`
    (d) Depends on compiler/OS.
    (e) Multiple types are tied for largest.
    (f) `long long`
    (g) `bool`
    (h) `short`

40. `cd ..` changes your current working directory to what directory?

    (a) The home directory
    (b) The directory that was previously used
    (c) The directory named ".."
    (d) The directory that contains the currently executing program
    (e) The parent directory

41. On my system, when I call `sizeof(bool);` I get the value `1`. What does this mean?

    (a) It means that a bool true value is equivalent to an integer 1.
    (b) It means that a bool takes up one byte.
    (c) It means that a bool's size is the same as the size of a function.
    (d) It means that a bool takes up one bit.
    (e) It means that a bool takes up one word (on a 64-bit machine this is 64 bits).
    (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.

42. For which values of `int x` will the following expression be true?
    `-2 < x <= 2`

    (a) -2, -1, 0, 1, 2
    (b) None exist
    (c) -2, -1, 0, 1
    (d) -1, 0, 1
    (e) All possible values of x
    (f) -1, 0, 1, 2

43. Which of the following is NOT a binary operator?

    (a) `++`
    (b) `+=`
    (c) `<<`
    (d) `>=`
    (e) `&&`
    (f) `=`
    (g) `/`

44. What is wrong with this function invocation?
    `std::cout << Func(int x, &y);`

    (a) `Func` is an invalid name due to the presence of an uppercase letter
    (b) The code will result in undefined behavior due to the lack of assignment
    (c) The `std::endl` was omitted
    (d) The ampersand (`&`) doesn't belong in a function call
    (e) Functions can't be invoked in larger expressions
    (f) The namespace for the function isn't specified
    (g) Arguments shouldn't have a type declaration

45. Accessing the value of an uninitialized `char` variable will result in what?

    (a) A random character from the ASCII chart
    (b) A compile-time error
    (c) A null pointer
    (d) Undefined behavior
    (e) A null character
    (f) A segmentation fault
    (g) The space character (`' '`)

46. What does the `std::endl` object do when passed as the second operand to the put to (<<) operator?

    (a) It doesn't do anything, it is instead used to indicate comments
    (b) It causes a newline character to be written to the stream
    (c) It causes the stream to separate words according to whitespace
    (d) It resets the stream
    (e) It can't be used with an insertion operator
    (f) It indicates the End-Of-File
    (g) It indicates that the stream should be prepared for new characters

47. What arguments is this function being called with?

```
int x = 4; int y = 17;
Func(++x, y--);
```

    (a) 4 and 16
    (b) 5 and 16
    (c) 4 and 17
    (d) 5 and 17

48. How do you make a variable that can have multiple types?

    (a) By using `static_cast`
    (b) By using `typedef`
    (c) By using `decltype`
    (d) By declaring its type as `auto`
    (e) It is impossible

49. What is the type of x?

```
auto x = My_Function("abcd");
```

    (a) `void`
    (b) `char`
    (c) `int`
    (d) `std::string`
    (e) Impossible to determine with the information given

50. 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

51. If a function is said to be overloaded, that means that there are two functions that share what property?

    (a) The same name
    (b) The same header file
    (c) The same body (using templates)
    (d) The same return type
    (e) The same parameters
    (f) The same arguments
    (g) The same libraries

52. Which of the following types are NOT fundamental types?

    (a) `char`
    (b) `unsigned`
    (c) `int`
    (d) `double`
    (e) `bool`
    (f) None of the above

53. What distinguishes the prefix increment operator from the postfix increment operator?

    (a) The prefix increment returns the value after the increment occurs.
    (b) The prefix increment operator is part of the C++ standard.
    (c) The prefix increment operator increments ints and double variables.
    (d) The prefix increment operator only increments non-zero values.
    (e) The prefix increment operator can be used in larger expressions.
    (f) None of the above

54. Curly braces in initialization (e.g. `int x {7};`), have what advantage over '=' initialization (e.g. `int x = 7;`)?

    (a) Constant initialization can occur.
    (b) The variable has local scope.
    (c) Compiler checks that initialization is zero initialized.
    (d) Implicit narrowing conversions are disallowed.
    (e) They are able to be used on user-defined types.
    (f) None of the above

55. When should you use type inference?

    (a) When you want to make the type clearly visible to readers.
    (b) When you want to be explicit about a variable's range or precision.
    (c) When the definition is in a large scope.
    (d) The variable has local scope.
    (e) When the type of the initializer isn't obvious.
    (f) None of the above

56. The `const` in an initialization implies what characteristic of the variable is constant?

    (a) Its location
    (b) Its type
    (c) Its representation
    (d) Its value
    (e) Its meaning
    (f) None of the above

57. Which of the following declarations follow the "East const Style"?

    (a) `Thing_const t;`
    (b) `const Thing t;`
    (c) `Thing t const;`
    (d) `Thing const t;`
    (e) `Thing t_const;`
    (f) None of the above

58. The "-i" flag when given to clang-format changes its behavior it what way?

    (a) It ensures that the formatted output is case-insensitive.
    (b) It allows for an interactive formatting where the edits can be reviewed by the user.
    (c) It allows for internal representations to be used, which results in more efficient code to be generated.
    (d) It does nothing in most cases, but causes the formatter to raise an error if the code is invalid C++.
    (e) It replaces the file with the formatted version.
    (f) None of the above

59. What is the result of the following expression? `7 % 5`

    (a) `0`
    (b) `1`
    (c) `2`
    (d) `3`
    (e) `4`
    (f) `5`
    (g) `6`
    (h) `7`

60. 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"?

    (a) `cd`
    (b) `cd ..`
    (c) `cd CSE232`
    (d) `cd Lectures\CSE232`
    (e) `cd ~`
    (f) `cd \`
    (g) None of the above

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

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

62. Why should you mark a variable as const?

    (a) To indicate to readers that the variable has a constant value
    (b) To add a compiler-enforced constraint on your code
    (c) To allow for more efficient compiled code to be generated
    (d) All of the above are true
    (e) None of the above are true

63. What is value of the expression `(’z’ - 1)`?

    (a) `’a’`
    (b) `’z`
    (c) `’!’`
    (d) `’y’`
    (e) `"z-1"`
    (f) `’Z’`

64. What distinguishes a void function from other functions?

    (a) It has no body
    (b) It hasn’t been declared
    (c) It has no parameters
    (d) It isn’t meant to be called
    (e) It doesn’t accept any input
    (f) It doesn’t write to output
    (g) It doesn’t return anything
    (h) It hasn’t been defined

65. If a file has a relative path of `../main/game/a.cpp`, what folder is that file found directly in?

    (a) `game`
    (b) the root directory
    (c) `main`
    (d) the home directory
    (e) `..`
    (f) the current working directory
    (g) All of the above

66. What does the `-i` mean when used with `clang-format`?

    (a) It means that case-insensitive changes should be performed
    (b) It means that intensive changes should be performed on non-meaningful whitespace
    (c) It means that indentation should be altered as needed to conform to the style guide
    (d) It means that the first version of the style guide should be used
    (e) It means that the following files should be used as input
    (f) It means the changes should directly change the file
    (g) None of the above are true

67. What are the benefits to using curly brace ({}) for initialization?

    (a) Avoidance of implicit narrowing conversions
    (b) Avoidance of the `auto` keyword
    (c) Avoidance of the most vexing parse
    (d) Avoidance of long type names
    (e) Avoidance of C-style casts
    (f) (a) and (b) are correct
    (g) (b) and (d) are correct
    (h) (a) and (c) are correct
    (i) All, except (e), are correct

68. For the following code, which of the following statements would result in an error when the program was executed?

```
int x = 0x12;
```

(a) x += 012;

(b) x++;

(c) cout << x;

(d) x = 4;

(e) int x2{x};

(f) None of the above would result in a runtime error.

69. Why does the following code give an error when compiling?

```
int x{0xAB};
double y = {1.23e3};
double z = -x + y;
z += x;
```

(a) You can't use assignment and initialization on the same variable

(b) The compound assignment operator can't be used on different types

(c) ints and doubles can't be combined with a binary operator

(d) 0xAB is not a valid int

(e) 1.23e3 is not a valid double

(f) -x isn't a legal operation

(g) None of the above are true

70. What does the following code output?

```
int x{4}, y;
y += (x++);
cout << y + x;
```

(a) 55

(b) 9

(c) 5+5

(d) 10

(e) 45

(f) 4+5

(g) Impossible to determine

71. Which of the following changes the type of the variable x from an int to a double?

(a) double const x = make_double(x);

(b) auto double x;

(c) double x = (double) x;

(d) x = static_cast<double>(x);

(e) None of the above

72. In the following code, what is the order of the function calls?

```
bark(3);
jump(lean());
hop(stop(), look(3));
```

(a) bark → lean → jump → stop → look → hop

(b) bark → lean → jump → look → stop → hop

(c) look → stop → hop → lean → jump → bark

(d) bark → lean → jump → stop → look → hop

(e) bark → jump → lean → hop → stop → look

(f) Impossible to determine with the information provided

73. If the following line compiles, which of the following are possible initializations of var?

```
cout << var + 1;
```

(a) int var5;

(b) double var = 3.4;

(c) int const var = 123'456;

(d) (a) and (b) are both possible

(e) (b) and (c) are both possible

(f) (a) and (c) are both possible

(g) All of the above are possible

74. When does the scope of a function parameter end?

    (a) When it is last accessed
    (b) When it is no longer needed
    (c) When the function is overloaded
    (d) When the function body ends
    (e) When it is last modified
    (f) When the first closing brace (}) is encountered
    (g) When the function is called

75. Which of the following is an example of Implementation Defined Behavior?

    (a) The value of uninitialized memory
    (b) The behavior of the unary plus operator
    (c) The scope of a variable
    (d) The size of certain fundamental types
    (e) The namespace of a function
    (f) The result of a static_cast
    (g) The order of evaluation of `h(x) + g(x)`

76. If you get the following compiler error, what is the cause of the problem?
`main.cpp:14:6: error: no member named ’cout’ in namespace ’std’`

    (a) The `#include <iostream>` was omitted
    (b) `std::` wasn’t placed before the `cout`
    (c) The using declaration (`using std::cout;`) was not included
    (d) Both (a) and (b) are the causes
    (e) Both (b) and (c) are the causes
    (f) Both (a) and (c) are the causes
    (g) All three are causes
    (h) None of the above are causes

77. For the following for loop, what would happen if the postfix increment was changed to a prefix increment?
`for (int i{0}; i++ < x; ) ...`

    (a) The value of `x` would be one larger in the body of the for loop
    (b) The loop would iterate one fewer time
    (c) The value of `x` would be one smaller in the body of the for loop
    (d) The final value of `x` would be one smaller
    (e) The final value of `x` would be one larger
    (f) The loop would iterate one more time
    (g) None of the above are true

78. What part of a variable is actually variable (able to change)?

    (a) Its type
    (b) Its representation
    (c) Its meaning
    (d) Its value
    (e) Its location
    (f) None of the above

79. What does the following code output?

```
double x{1/3};
cout << x;
```

    (a) 0.33333334
    (b) 0
    (c) 1
    (d) .33
    (e) 1/3
    (f) None of the above

80. Which of the following operators is equivalent to an ”AND” operation in logic?

    (a) &&
    (b) ^
    (c) &
    (d) |
    (e) ||
    (f) Two of the above are correct

81. What Linux command is used to display the contents (files/folders) of the current working directory?

   (a) `ls`
   (b) `print`
   (c) `g++`
   (d) `pwd`
   (e) `code`
   (f) `cd`
   (g) `clang-format`
   (h) `show`

82. In order to autocomplete the name of a file/folder, what character should be typed into the terminal?

   (a) `tab`
   (b) `Right Arrow`
   (c) `enter`
   (d) `space`
   (e) `Left Arrow`
   (f) `q`
   (g) `Control-f`
   (h) `Control-c`

83. The textbook makes heavy use of the `auto` keyword, and to be fair, its explanation of the circumstances where it should be used is true. However, why does CSE 232 caution against its use in this course?

   (a) `auto` is permitted in practice, but since the autograder can't support it, it can't be used on assignments
   (b) `auto` can conceal type errors that make debugging more difficult
   (c) `auto` rarely is useful beyond simple examples, so it isn't worth learning
   (d) Although `auto` is indeed useful, few compilers support it
   (e) `auto` fills the scope with many unnecessary names, which slows compilation time
   (f) It is easy to become dependant on `auto` to avoid making explicit decisions about type
   (g) CSE 232 does actually encourage the use of `auto`

84. 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 strings.
   (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.

85. What is the value of `x`?

```
int x{1234 % 5};
```

   (a) 0
   (b) 1
   (c) 4
   (d) 5
   (e) 123
   (f) 1230
   (g) 1234

86. Which of the following statements will cause 232 to be written to standard output?

   (a) `cout << 231 + 1;`
   (b) `cout << 23 << 2;`
   (c) `cout << "2" << '32';`
   (d) `cout << '2' << 32;`
   (e) `All`, except (a)
   (f) `All`, except (b)
   (g) `All`, except (c)
   (h) `All`, except (d)

87. Why must you always initialize const variables?

    (a) Because const variables have a namespace scope
    (b) Because they are dangerous to use when uninitialized
    (c) Because const variables' size is implementation specific behavior
    (d) Because const values can only be used after they are out of scope
    (e) Because they can't be assigned a value afterwards
    (f) Wrong, you aren't required to initialize const variables

88. What is the name of the directory that contains all of the files and folders on a computer?

    (a) Parent
    (b) Working
    (c) Base
    (d) Root
    (e) Top
    (f) Home
    (g) None of the above

This page intentionally left blank.