Sample Midterm Exam
CSE 232 (Introduction to Programming II)
Fall 2019

VERSION 1

Full Name: ............................................................................................................

PID: ......................................................................................................................

Instructions:
• DO NOT START/OPEN THE EXAM UNTIL TOLD TO DO SO.
• You may however write and bubble in your name, PID and form number (with a #2 pencil) on the front of the printed exam and bubble sheet prior to the exam start.
• Present your MSU ID (or other photo ID) when returning your bubble sheet and printed exam.
• Only choose one option for each question.
• Assume any needed #includes and using std::...; namespace declarations are performed for the code samples.
• Every question is worth 2 points. There are 55 questions in the real exam. The exam is worth 100 points. Thus you get 5 or fewer questions wrong, you will get full credit.
• 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, please raise your hand and a proctor will assist you.

Figure 1: http://xkcd.com/499/
1. Who invented C++?
   (a) Abdol Esfahanian
   (b) Bjarne Stroustrup
   (c) Charles Owen
   (d) IBM
   (e) C++ itself
   (f) Guido van Rossum
   (g) Josh Nahum
   (h) None of the above

2. How long is the regrade request period for this course?
   (a) 1 day
   (b) 4 days
   (c) 1 week
   (d) 4 weeks
   (e) Until the last day of class
   (f) Forever

3. If you got a zero on the midterm, how many points would you need on the final to pass the class?
   (a) It is impossible to pass the class with a zero on the midterm.
   (b) 100 of 200
   (c) 150 of 200
   (d) 0 of 200
   (e) None of the above

4. What is the name of the ampersand (&) operator?
   (a) The address-of operator
   (b) The and operator
   (c) The dereference operator
   (d) The pointer operator
   (e) None of the above

5. How large is an int?
   (a) 1 byte
   (b) 2 bytes
   (c) 4 bytes
   (d) 8 bytes
   (e) Depends on the system

6. When is a pointer better than a reference?
   (a) When you need multiple names for the same variable.
   (b) When a copy needs to be avoided.
   (c) When you are writing C++17 code.
   (d) None of the above.

7. What is x in this declaration?
   const string * x;
   (a) A pointer to a string
   (b) A pointer to a constant string
   (c) Syntax Error
   (d) A constant pointer to a constant string
   (e) A constant pointer to a string
   (f) None of the above

8. Which of the following cause a string to be copied (presuming y is a string)?
   (a) const string * x = y;
   (b) const string & x = y;
   (c) const string * const x = y;
   (d) string & x = y;
   (e) string * x = y;
   (f) None of the above

9. How many labs can be missed without a grade penalty?
   (a) 4
   (b) 3
   (c) 2
   (d) 1
   (e) 0

10. Which of the following is a runtime error?
    (a) Redeclaring a variable
    (b) Assigning a string to an int variable
    (c) Forgetting a semicolon (;)
    (d) Casting an int to a double
    (e) Using postfix increment when prefix was needed
11. What is the best reason to obey a style guide?
   (a) Because it makes your code easier to write
   (b) Because it makes your code easier to debug
   (c) Because it makes your code faster to run
   (d) Because it makes your code easier to read
   (e) Because it makes your code faster to compile

12. What is NOT a benefit of statically typed languages (like C++)?
   (a) The code is more flexible
   (b) The compiler can check for type correctness
   (c) It allows the code to be optimized for speed
   (d) The programmer is explicit about the type of each variable
   (e) None of the above

13. What is NOT included when initializing a variable?
   (a) The variable’s name
   (b) The variable’s type
   (c) The variable’s value
   (d) None of the above

14. Can you declare (without initializing) a reference?
   (a) No
   (b) Only if the reference is const
   (c) Only if the reference is for a fundamental type
   (d) Yes

15. What line number is causing the syntax error in:
    02.1-numericTypes.cpp:8:39:
    error: expected ')'?
   (a) 2
   (b) 39
   (c) 8
   (d) 1

16. Why is it important to frequently compile and test your code?
   (a) To prevent compiler errors from becoming runtime errors
   (b) To ensure that your work is being saved
   (c) To enable the compiler to optimize your code
   (d) To reduce the time between adding a bug and finding it

17. Which of the following is NOT a preprocessor statement?
   (a) #define SOME_NAME
   (b) #ifndef SOME_NAME
   (c) #pragma once
   (d) #include <iostream>
   (e) All of the above are preprocessor statements

18. How many different kinds of comments are there in C++?
   (a) There are no comments in C++
   (b) 1 kind
   (c) 2 kinds
   (d) 3 kinds

19. What is the name of the :: operator?
   (a) The Within Operator
   (b) The Namespace Operator
   (c) The Scope Resolution Operator
   (d) None of the above
20. What is wrong with the following line?
   string x = 'c';

   (a) Strings cannot be initialized.
   (b) There is a name error.
   (c) There is a syntax error.
   (d) There is a type error.
   (e) Nothing is wrong.

21. Is the variable name milesPerHour allowed?

   (a) Yes, it is legal and correctly styled.
   (b) No, it violates the style guide.
   (c) No, it is an illegal name.
   (d) No, variable names shouldn’t include full words.
   (e) None of the above are true.

22. What does it mean to have an overloaded operator?

   (a) It means the operator’s behaviour is undefined under certain circumstances.
   (b) It means the operator can be used with multiple types.
   (c) It means the operator is used by the iostream library to output values.
   (d) It means the operator is heavily used.
   (e) None of the above.

23. Which of the following namespace techniques is disallowed by this course?

   (a) using std::cout; cout << "Hi";
   (b) std::cout << "Hi";
   (c) using namespace std; cout << "Hi";
   (d) None of the above.

24. Why is a full namespace merge a bad idea?

   (a) Because it forces the compilation of the entire STL
   (b) Because it makes it difficult to determine where a name came from.
   (c) Because it slows down programmer productivity by requiring more inventive names
   (d) Because it makes compilation slower
   (e) All of the above.

25. If you declare a string, what is its initial value?

   (a) false
   (b) Empty
   (c) 0
   (d) Undefined
   (e) None of the above.

26. If you declare a int, what is its initial value?

   (a) Empty
   (b) 0
   (c) Undefined
   (d) false
   (e) None of the above.

27. Which of the following types is largest?

   (a) bool
   (b) char
   (c) short
   (d) int
   (e) long
   (f) long long
   (g) Multiple types are tied for largest.
   (h) Depends on compiler/OS.
28. Which of the following statements about `int` and `long` is true?

(a) `int` is larger than `long`
(b) `int` is smaller than `long`
(c) `int` is smaller or equal in size to `long`
(d) `int` is larger or equal in size to `long`
(e) None of the above

29. Is `'9' + 1'; legal?

(a) Yes, because both operands are digits.
(b) No, because addition is not an expression.
(c) No, because you can’t add a char to an int.
(d) Yes, because both operands are integral types.
(e) No, because you can’t add a letter to a digit.

30. Which of the following is NOT a legal initialization?

(a) `int x = 0xA;`
(b) `int x = 1;`
(c) `int x = {1};`
(d) `int x = 01;`
(e) `int x(1);`
(f) `int x; x = 1;`

31. Why are copies generally bad?

(a) Because they require more memory.
(b) Because copying takes additional time.
(c) Because they limit the ability to modify the original variable.
(d) All of the above.

32. What is the result of `3 / 4.0`?

(a) 0.75
(b) 0
(c) 1
(d) None of the above

33. What is the result of `(cin >> x)`?

(a) `x`
(b) `>>`
(c) `cin`
(d) Depends on if the insertion was successful.

34. What is the result of `static_cast<long>(10.7)`?

(a) 10
(b) 10.7
(c) 11
(d) Depends on if the long is signed.

35. What is the result of `(3 / 4)`?

(a) 0.75
(b) 1
(c) 0
(d) Depends on the variable it is assigned to.

36. Which of the following operators have side-effects?

(a) The insertion operator.
(b) The extraction operator.
(c) The assignment operator.
(d) The postfix increment operator.
(e) All of the above.

37. Which is `x` after `int y = 8; int x = y++;`?

(a) 8
(b) 7
(c) 9
(d) Undefined
(e) y
(f) None of the above.

38. Which is the result of `(true + 4)`?

(a) 4
(b) 5
(c) Undefined
(d) Syntax Error
(e) None of the above.
39. Which is the result of:
   int x = 4, y = 0;
   if ((x = 0) && (y = 10)) {...}

   (a) Undefined
   (b) x is 4 and y is 0
   (c) x is 4 and y is 10
   (d) Syntax Error
   (e) x is 0 and y is 10
   (f) x is 0 and y is 0
   (g) None of the above.

40. Which is b after:
   int x = -7;
   bool b = (0 <= x <= 9);

   (a) -7
   (b) true
   (c) Syntax Error
   (d) false
   (e) Undefined
   (f) None of the above.

41. How many statements compose the body of an if statement?

   (a) 0
   (b) 1 or more
   (c) 0 or more
   (d) 1 (but it must be a block statement)
   (e) 1
   (f) None of the above.

42. What is printed by the following code?
   int x = 3;
   if (!x)
   if (x = 4)
      cout << "AAA";
   else
      cout << "BBB";

   (a) Compile-time error
   (b) BBB
   (c) No output generated
   (d) BBBAAA
   (e) AAABBB
   (f) AAA
   (g) None of the above

43. What is the output from the following code:
   int x = 3;
   do
      cout << x;
   while (x--);

   (a) 3210
   (b) 333
   (c) 33333
   (d) No output generated
   (e) Compile-time error
   (f) 3210-1
   (g) 3333
   (h) 321
   (i) None of the above.

44. Which of the following is NOT a legal infinite loop?

   (a) for (;;)
      {...}
   (b) for (;-1;)
      {...}
   (c) while (7)
      {...}
   (d) while (True)
      {...}
   (e) All of the above are legal infinite loops

45. For loops have three parts (here named a, b, c, d in for (a; b; c) {d}). For a loop that iterates twice over its body, what is the order of execution of a, b, c and d?

   (a) a → b → d → c → b → d → c → b → d → c
   (b) a → b → d → c → b → d → c → b → d
   (c) a → b → c → d → a → b → c → d
   (d) a → b → d → c → b → d → c → b
   (e) a → b → d → c → b → d
   (f) None of the above
46. Assuming s is a string, which of the following statements results in an infinite loop?
   (a) for(auto i=s.size()-1; i>0; i--)
   (b) for(auto i=0; i<s.size(); i++)
   (c) for(auto i=s.size()-1; i>=0; i--)
   (d) for(auto i=0; i<s.size()-1; i++)
   (e) All of the above
   (f) None of the above

47. What does this declaration declare?
   ```
   string* x, y, z;
   ```
   (a) Run-time Error
   (b) Compile-time Error
   (c) x and y are pointers to a string, z is a string
   (d) x is a pointer to a string, y is a string, z is a string
   (e) x, y and z are pointers to string types
   (f) None of the above

48. Which of the following is illegal?
   (a) int *ip;
   (b) string s, *sp = nullptr;
   (c) int i; double* dp = &i;
   (d) int *pi = 0;
   (e) All of the above
   (f) None of the above

49. What will happen in this code?
   ```
   int a = 100, b = 200;
   int *p = &a, *q = &b;
   p = q;
   ```
   (a) a is assigned to b
   (b) b is assigned to a
   (c) Run-time error
   (d) Compile-time error
   (e) p now points to a
   (f) p now points to b
   (g) a and b have been swapped
   (h) q now points to a
   (i) None of the above

50. The correct statement for a function declaration that takes a const pointer to a string that needs to be revised (in the body of this function, the pointer will be be used to modify its pointed at string, but cannot be pointed at a different object), a pointer to an int, and returns a reference to an int is:
   (a) & int fun(const * string const, * int)
   (b) int & fun(const string *, int *)
   (c) int & fun(const string * const, int *)
   (d) int & fun(string * const, int *)
   (e) & int fun(const * string, * int)
   (f) Multiple answers are correct
51. What is the output of this program?
```cpp
void func(int &a, int &b) {
    int c = a;
    a = b;
    b = c;
    cout << "In func " << a
         << b;
}
int main() {
    int a = 5, b = 10;
    func(a, b);
    std::cout << " In main " << a
              << b;
    return 0;
}
```
(a) In func 10
(b) In func 10 In main 105
(c) Run-time error
(d) Compile-time error
(e) In func 5 In main 510
(f) In func 5 In main 105
(g) None of the above

52. What is the output of this program?
```cpp
int a = 9;
int &aref = a;
aref++;
std::cout << a;
```
(a) 10
(b) 11
(c) 12
(d) Compile-time error
(e) 9
(f) Run-time error
(g) None of the above

53. What is the output of this program?
```cpp
void func (int *b) {
    *b = 1;
}
int main () {
    int *a;
    int n;
    a = &n;
    *a = 0;
    func(a);
    cout << *a << endl;
}
```
(a) Compile-time error
(b) 1
(c) The address of b
(d) The address of n
(e) Run-time error
(f) The address of a
(g) 0
(h) None of the above

54. What is the output of this program?
```cpp
int a = 9;
int *aptr = &a;
aptr++;
aptr++;
std::cout << a;
```
(a) 9
(b) 11
(c) Run-time error
(d) Compile-time error
(e) 10
(f) 12
(g) None of the above
55. What is the output of this program?

```cpp
void wth(int i, int &k) {
    i = 1;
    k = 2;
}

int main () {
    int x = 0;
    wth(x, x);
    cout << x << endl;
    return 0;
}
```

(a) 2  
(b) wth  
(c) 1  
(d) Run-time error  
(e) 0  
(f) Compile-time error  
(g) None of the above

56. After the following statements, which option changes the value of i to 143?

```cpp
int *p;
int i, k;
i = 142;
k = i;
p = &i;
```

(a) k = 143;  
(b) *k = 143;  
(c) p = 143;  
(d) *p = 143;  
(e) Both (a) and (c)  
(f) Both (a) and (d)  
(g) None of the above

57. Can two functions have the same name?

(a) Yes, but only if they have different names of parameters.  
(b) Yes, but only if they have different number and/or types of parameters.  
(c) Yes, but only if they have a different number of parameters.  
(d) No, C++ doesn’t support function overloading.  
(e) No, only methods and operators can have the same names, not functions.  
(f) None of the above

58. What is a recursive function?

(a) A function that returns the same value, if called with the same arguments.  
(b) A function that doesn’t use loops.  
(c) An iterable function.  
(d) A function that can be used with different number of arguments.  
(e) A function that calls itself.  
(f) None of the above

59. Which of the following is NOT a integral (integer-like) literals?

(a) 12  
(b) ’3’  
(c) 0x34a  
(d) 0  
(e) 054  
(f) All of the above are integral types
60. Which of the following is a **NOT** good reason to use the unsigned integer type in C++?

(a) Because a library function returns it.
(b) Because you will need to perform bitwise operations.
(c) Because you need the expected overflow/underflow behaviour.
(d) Because a value can never be negative.
(e) All of the above are valid
(f) None of the above are valid

61. What is the type of `x` in:

```cpp
string str = "hi";
auto x = str.size();
```

(a) `string::size_type`
(b) `unsigned int`
(c) `long`
(d) `3`
(e) `int`
(f) `2`
(g) None of the above

62. When will the string’s length and size methods return different results?

(a) When the string hold UTF8 characters
(b) When the string has been initialized
(c) When the string has been appended to
(d) When the string is empty
(e) When the strings capacity is different from its size
(f) When the string has been passed by reference to a function
(g) Never, they always are the same

63. When should you use `x.at(1)` instead of `x[1]` to access the second character of the string `x`?

(a) When you need to be able to assigned to the resulting rvalue.
(b) When you need the fastest performance.
(c) They are the same, so either is always fine.
(d) When you need an error if `x` is shorter than two characters.
(e) None of the above.

64. When should you use `string::npos`?

(a) It is needed to convert strings to ints.
(b) To indicate that you want to refer to past the end of a string.
(c) When every you need the largest positive value an int can hold.
(d) When you need a very large integer.
(e) None of the above.

65. Does C++ support default values for function arguments?

(a) Yes, but they must be specified in the implementation file.
(b) No, all parameters must be matched to arguments.
(c) No, but operator overloading can achieve the same effect.
(d) Yes, but they must be specified in the header file.
66. What is the purpose of header guards?
   (a) To allow for optimized compilation speeds.
   (b) To stop private functions from being accessed publicly.
   (c) To ensure that all headers are included by the needed implementation files.
   (d) To ensure that every function is declared prior to invocation.
   (e) To allow for default argument values and function overloading.
   (f) To prevent re-declaration by ensuring a header is only included once.

67. What will occur if two header guards use the same variable name?
   (a) Only one of the two headers will be included.
   (b) Both headers will be included.
   (c) Neither header will be included.
   (d) Either (a), (b), or (c) will randomly occur.
   (e) None of the above.

68. Which of the following Unix command line commands will copy a file?
   (a) c
   (b) copy_file
   (c) cp
   (d) cat
   (e) copy
   (f) cp_file
   (g) mv

69. At the command line, the Unix command "cd .." does what?
   (a) Prints the working directory
   (b) Changes the permissions of the current directory
   (c) Copies the contents of the current working directory into the child’s directory
   (d) Copies all of the children of the current directory into the cd directory
   (e) Changes the current working directory to be the parent directory.

70. Which of the following are true about templated functions?
   (a) It is itself not a function, but a way to create a function
   (b) It contains the keyword template
   (c) It makes use of a template parameter to represent a calling type.
   (d) All of the above
   (e) None of the above