03-60-340-30
2017 Winter, Tues. Feb. 7, 2017 in TC 204
University of Windsor, School of Computer Science

Midterm 1 Examination Solutions

Mr. Paul Preney

Student ID:	
FIRST Name:	
LAST Name:	

"I have neither given nor received unauthorized help with this examination. Any suspicion of cheating will automatically void my mark on this examination."

Signature
Unsigned examination booklets will not be graded.
Signature implies agreement with the above statement in quotes.

INSTRUCTIONS

1. You have **1 hour** maximum to complete this examination. Pace yourself accordingly.
2. Write your answers in the space provided. No additional space will be provided.
3. Do **not** remove any papers from this booklet or add new ones.
4. You may **not** use any reference material(s) **except** what has been provided within this examination booklet and the course text books *The C++ Programming Language, 4th Edition* and/or *The C++ Standard Library: A Tutorial and Reference*.
5. **You may not use the C Standard Library unless given explicit permission to do so.** C++ coding techniques and the C++ Standard Library without the C Standard Library subset must always be used.
6. **Document your code where appropriate.** Unclear answers may not receive partial marks. Ensure any written English uses proper spelling, grammar, and can be understood. Answers must be neat and legible to receive marks.
7. **Be sure** that you have printed your name and student number on all pages of this examination.
8. Ensure that you have all **10 pages** of this examination (including this page) before starting to write this exam. If you don't, bring this to the attention of the instructor immediately.
9. Ensure the proper case, spelling, syntax, grammar, and punctuation marks are correctly used in all answers involving code.

EXAMINATION MARK: ______________

MAXIMUM MARK: 73

Part I: Multiple Choice and Short Answer Questions (45 marks)

For each question in this section, neatly and plainly **circle or underline** the **single** response which most correctly completes/answers the statement/question given for multiple choice or True/False questions, otherwise, write in the appropriate answer(s) in the space provided. Read carefully! Unintelligible or ambiguous responses will receive a mark of zero (0) for that question, so ensure that your answer is clear.

Q1) True/False: Strictly speaking and unlike ISO C, ISO C++ only supports pass-by-value when calling functions. **[1 mark]**

(a) True [✓ (b) False]

Q2) The **first** official ISO C++ standard is associated with the year ____.

Answer: ___1998____ **[1 mark]**

Q3) To compile **C++14** code using GCC's g++, the compiler option one must use is: ______.

Answer: ___-std=c++14____ **[1 mark]**

Q4) True/False: In C++, one can declare a variable whose type is a reference to a reference. **[1 mark]**

(a) True [✓ (b) False]

Q5) The creator of C++ originally created the language while working at ______. (Provide company name.)

Answer: ___Bell labs____ **[1 mark]**

Q6) All C++ types that have as their rightmost decoration one or two ampersands are known as ______ types.

Answer: ___reference____ **[1 mark]**

Q7) If a C++ compiler chose to implement a variable of type **char const*&** as a pointer, what would the pointer variable's (exact equivalent) type be?

Answer: ___char const * * const____ **[2 marks]**

Q8) If a C++ compiler chose to implement a variable of type **Foo&** as a pointer, what would the pointer variable's (exact equivalent) type be?

Answer: ___Foo * const____ **[2 marks]**

Q9) If a C++ compiler chose to implement a variable of type **Bar*&&** as a pointer, what would the pointer variable's (exact equivalent) type be?

Answer: ___Bar * * const____ **[2 marks]**

Q10) Given the declaration, `int i;`, what does `i++;` return? **[1 mark]**

(a) an l-value [✓ (b) an r-value] (c) neither of these

Q11) Write a C++ expression that declares a reference to a temporary int object who's value is 5.

Answer: ___int&& r = 5;____ **[1 mark]**

Q12) A variable can be easily determined to be an l-value when it ______.

Answer: ___has a name____ **[1 mark]**

Q13) A variable can be easily determined to be an r-value when it _____.

Answer: ___doesn't have a name____ **[1 mark]**

Q14) True/False: ISO C++'s evolution is driven by theoretical problems not by real ones. **[1 mark]**

(a) True [✓ (b) False]

Q15) True/False: C++ permits implicit violations of the static type system. **[1 mark]**

(a) True [✓ (b) False]

Q16) Match the appropriate term with the best definition. The possible terms to choose from are:

imperative, modular, object-based, object-oriented, generic, functional **[6 marks]**

Definition	**Write Term Associated With Definition In Row**
Computation is defined in terms of patterns with placeholders that act on common aspects of those placeholder's interfaces.	generic
Computation is defined in terms of state that determines the expression of the next operation to be performed.	object-based
Computation is defined in terms of programming statements that describe changes in state.	imperative
Computation is defined in terms of hierarchically organized state that determines the expression of the next operation to be performed.	object-oriented
Computation is defined in terms of operations that have no side effects.	functional
Computation is defined in terms of programming statements that describe changes in state with the additional ability to expose/hide functions and/or variables as seen from other translation units or namespaces.	modular

Q17) Per ISO C++ rules, to use C's <ctype.h> header in a C++ program, **explain** what one has to do to #include the header in C++. Be sure to also write the final C++ #include code for <ctype.h> in your answer. **[3 marks]**

___1) delete the '.h' suffix from the header name____

___2) prefix a 'c' to the header name____

___e.g., <ctype.h> => <cctype>____

____

Q18) All C Standard Library functions are contained in the **cstd** namespace. **[1 mark]**

(a) True [✓ (b) False]

Q19) All C++ Standard Library functions are contained in the **cxxstd** namespace. **[1 mark]**

(a) True [✓ (b) False]

Q20) Briefly, what is the **difference** between **cout** and **cerr**? **[2 marks]**

___cout outputs to standard output____

___cerr outputs to standard error; cerr is is unbuffered_____________________________________

____

____

Q21) The Standard Library uses the ____ bit shift operator to write formatted data out to an ostream.

Answer: ___<<____ **[1 mark]**

Q22) The Standard Library uses the ____ bit shift operator to read in formatted data from an istream.

Answer: ___>>____ **[1 mark]**

Q23) Originally in C++, the **auto** keyword prefixed in front of a variable declaration in a function meant that the variable was declared and placed on a function's _____.

Answer: ___call stack____ **[1 mark]**

Q24) In the current C++ standard, using the **auto** keyword to declare a variable allows a programmer to do what in C++? **[1 mark]**

___e.g., one of:____

___1) omit the type when declaring a variable______________________________________

___2) omit the return type of a function____

____

Q25) Using the current C++ standard, will writing **auto i;** in **main()** compile? Answer yes or no. If no, explain why it will not compile. **[2 marks]**

____No. The compiler must be able to determine the exact type of i, e.g., via______________

____constructor syntax, code to the right of the '=' after the variable, etc._______________

Q26) Rewrite the function **double foo(int i) { return 3.14; }** to use the new C++ function suffix declarator syntax. (Hint: Think "suffix" and know you need to use -> .) **[2 marks]**

____auto foo(int i) -> **double** { return 3.14; }____________________

Q27) Rewrite the function **double foo(int c) { return 3; }** so that it is a C++ lambda function. (Don't over-complicate this!) **[2 marks]**

____[](int c) -> **double** { return 3; }____________________

____NOTE: Unless 3 is changed to 3.0 the return type, i.e., -> double, must be present._____

Q28) If a C++ I/O Stream type is cast to a **bool**, what do the **true** and **false** results mean? **[3 marks]**

____true means no errors or EOF have occurred on the stream;____________________

____false means an error or EOF has occurred on the stream.____________________

Q29) True/False: The ISO C++ standards dictate complexity requirements for nearly everything in the C++ Standard Library. **[1 mark]**

[✓ (a) True] (b) False

Part II: General Questions (28 Marks)

Answer all parts of each question in the space provided below each question. The number of marks assigned to each question is indicated at the end of each question. You are expected to answer questions using complete sentences and proper grammar. If the answer is program code, simply write the code fragment that answers the question **unless you are explicitly asked to write a full-and-complete program**.

Q30) Write the code fragment to declare a variable called **values** whose type is a **std::vector** containing **std::string** instances. **[2 marks]**

```
std::vector<std::string> values;
```

Q31) Write the code fragment to read in all **std::string** words from **std::cin** (using a single **while** loop) into the variable **values** whose declaration you wrote in Q30. (An EOF or any occurrence of an error constitutes the end of the words to be read in.) Remember to declare your **std::string** variable. **[4 marks]**

```
std::string word;
while (std::cin >> word)
      values.push_back(word);
```

Q32) Use the C++ **range-based for-loop** to output all words in the **values** variable (from Q30 and Q31) to **std::cout**. **[3 marks]**

```
for (auto i : values)
      std::cout << i << '\n';
```

Q33) Call **<algorithm>'s std::for_each** function passing the arguments required to output all words in the **values** variable (from Q30 and Q31) to **std::cout** within a single C++ statement. Recall the prototype for for_each is:

```
template <typename Iter, typename UnaryPredicate>
UnaryPredicate for_each(Iter first, Iter last, UnaryPredicate pred);
```

[4 marks]

```
std::for_each(
      values.begin(),
      values.end(),
      [](auto e)
      {
            std::cout << e;
      }
);
```

Q34) Given the **values** variable (from Q30 and Q31) write the complete code fragment that performs the following:

- Starting at the beginning of **values**, search for the word "**the**" using **<algorithm>'s std::find**. Store the result in a variable called "**the_pos**".
    - NOTE: This must be done using a single C++ statement.
- Starting at "**the_pos**", search for the first word whose string length (i.e., string_var.size()) is greater than 10 using <algorithms>'s **std::find_if**. Store the result in a variable called "**big_pos**".
    - NOTE: This must be done using a single C++ statement.
- As you did in Assignment 1, declare a **std::map** capable of representing a frequency histogram of words. Call this variable "**hist**".
    - NOTE: This must be done using a single C++ statement.
- Using **<algorithm>'s std::for_each**, add everything in the range **[the_pos, big_pos)** to the histogram **hist** variable.
    - NOTE: This must be done using a single C++ statement.
    - Hint 1: Recall how the histogram was constructed in Assignment 1 which included using [].
    - Hint 2: Use a lambda function to add the data to the histogram.
    - Hint 3: Remember to capture something!

The prototypes for **std::find**, **std::find_if** are as follows:

```
template <typename Iter, typename T>
Iter find(Iter first, Iter last, T const& value);

template <typename Iter, typename UnaryPredicate>
Iter find_if(Iter first, Iter last, UnaryPredicate pred);
```

[15 marks]

```
auto the_pos = std::find(begin(values), end(values), “the”);

auto big_pos = std::find_if(
      the_pos, end(values),
      [](auto str) { return str.size() > 10; }
);

std::map<std::string, unsigned> hist;

std::for_each(
      the_pos, big_pos,
      [&hist](auto str) mutable { ++hist[str]; }
);
```

This page was intentionally left blank. You may write answers on it.

This page was intentionally left blank. You may write answers on it.