03-60-340-01
2013 Winter, Tues. Jan. 29, 2013 in BB 113
University of Windsor, School of Computer Science

Midterm 1 Examination Sample 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 **45 minutes** 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 book *C++ In A Nutshell*.
5. **You may not use the C Standard Library unless given explicit permission to do so.** This is a course on C++ --not C. C++ coding techniques and the C++ Standard Library without the C Standard Library subset must always be used. If you have any questions concerning this, then ask for clarification.
6. **Document your code where appropriate.** Unclear code may not receive partial marks without documentation. 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 **8 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: 54

Part I: Multiple Choice and Short Answer Questions (49 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) The C++ programming language was created by _____ (full name).

Answer: ___Bjarne Stroustrup_____ **[1 mark]**

Q2) C++ was originally called _____.

Answer: ___C with Classes_____ **[1 mark]**

Q3) Briefly explain what the ISO C++'s committee "zero overhead rule" means in terms of design. **[1 marks]**

___If you don't use a feature you don't pay for it (in terms of run-time overheads/costs)._____

_____

Q4) Explain the key differences between (i) modular and object-based, and, (ii) object-based and object-oriented programming. **[4 marks]**

(i) ___Modular has only one instance of the module whereas in object-based the module is the___

___object and therefore one can have multiple instances._____

_____

(ii) ___Object-based programming does not allow/permit one to use inheritance; object-oriented___

___programming does._____

_____

Q5) C++ is a multi-_____ programming language.

Answer: ___paradigm_____ **[1 mark]**

Q6) Briefly explain **why** the C programming language can **only** be said to **truly** support pass-by-value. **[2 marks]**

___C does not support references. Although C does support pointers, they are always copied____

___when passed to functions and the programmer must also explicitly specify the address as a___

___value when passing that argument. Thus, each pointer argument is pass-by-value._____

_____

Q7) C++11 supports two types of references. What are they called?

Answer A: ___L-value references (i.e., & as right-most symbol in a type declaration)_____ **[1 mark]**

Answer B: ___R-value references (i.e., && as right-most symbol in a type declaration)____ **[1 mark]**

Q8) Clearly explain what the differences, if any, are between `T *const` and `T const*`. **[2 marks]**

___T *const is a constant (read-only) pointer to a non-constant (read/write) value of type T._____

___T const* is a non-constant (read/write) pointer to a constant (read-only) value of type T._____

_____

_____

Q9) If the reference to some type, `T`, where to be written as `T&`, what would the semantically equivalent pointer type declaration to that reference be written as?

Answer: ___T * const *_____ **[2 marks]**

Q10) Using big-O, little-o, and omega complexity symbols, explain what the **cost of moving data** is in terms of **copying data** and **copying pointers to the data**. Also, is the cost of **moving data** ever zero? **[3 marks]**

___O(moving data) = o(copying data), i.e., moving is no worse than copying_____

___Ω(moving data) = Ω(copying pointers to the data), i.e., moving is at least copying pointers __

___Ω(moving data) is never zero, i.e., moving is never free of cost (clearly assuming that there__

___some data to actually move!)._____

_____

_____

Q11) One should view a _____ [1] operation as an optimized ____ [2] operation. (Hint: Q10.)

Answer 1: ___moving_____ **[1 mark]**

Answer 2: ___copying_____ **[1 mark]**

Q12) Write a C++11 lambda function that accepts an `int` as an argument and returns twice its value (as an `int`). **[2 marks]**

```
[](int i)
{
   return i*2;
}
```

Q13) The mathematician that designed key portions of the STL is _____ (full name).

Answer: ___Alexander Stepanov_____ **[1 mark]**

Q14) Briefly describe what a C++ Standard Library container represents. **[1 mark]**

___A container is an entity / a data structure that can hold many values all of the same type._____

_____

_____

Q15) Briefly describe what a C++ Standard Library iterator represents. **[1 mark]**

___An iterator represents a pointer/cursor to a specific element value inside a container._____

___An iterator also has a well-defined traversal order to transit from element to element._____

_____

Q16) C++ Standard Library's iterators were modeled upon which C language construct?

Answer: ___pointers_____ **[1 mark]**

Q17) In C++, object-oriented programming provides run-time polymorphism whereas its _____ programming provides compile-time polymorphism.

Answer: ___generic (template)_____ **[1 mark]**

Q18) Briefly explain what is meant by a predicate in the C++ Standard Library. **[1 mark]**

___A predicate is a stateless function that always returns a specific bool value based on its_____

___arguments._____

Q19) True or false: In C++ a Standard Library function requiring a predicate allows the predicate to be stateful. **[1 mark]**

(a) True (b) False

Q20) In the C++ Standard Library, all sorting operations rely on a _____ (3 words) to sort things. The default overloaded operator to perform such is _____.

3-word answer: ___strict weak order_____ **[1 mark]**

Operator answer: ___< (i.e., less than)_____ **[1 mark]**

Q21) Why is C++'s use of the three-word answer with only a single operator (i.e., in Q20) better to use for sorting than using two operators. Briefly explain. **[2 marks]**

___Using one operator ensures consistent semantics for the equivalence and less than_____
___operations since both are defined in terms of the one operator. If two operators are used, it is_
___more easily possible for their return values to violate the ordering semantics required for_____
___the sort and therefore result in an invalid sort / corrupted data structure._____

Q22) Why is C++'s use of the three-word answer with only a single operator (i.e., in Q20) enable one to do that cannot be done with a total order? Briefly explain. **[2 marks]**

___Strict weak order (SWO) semantics allow both (a < b) and (b < a) to both return false. This__
___is not possible with total order semantics since (a < b) => not (b < a). In effect, a SWO_____
___permits one to compare things that cannot be compared. Since equivalence is defined to be___
___!(a < b) && !(b < a), this places all incomparable elements into the same equivalence class.___

Q23) Briefly explain what the differences between a forward iterator and a bidirectional iterator are in terms of their permitted operations. **[1 mark]**

___A bidirectional iterator also permits traversing in reverse order (from any position) using the__
___decrement operators (prefix and postfix) in addition to the in-order traversal of forward iters.__

Q24) Clearly describe what the C++ compiler adds to a C++ class/struct when one declares a virtual member function in it. Briefly what allows the correct virtual member function to be correctly invoked from all derived classes. **[4 marks]**

___The C++ compiler will internally add a function pointer for each virtual member function.___
___These function pointers are typically placed into an array pointed to by a internally added_____
___pointer in the class/struct since this array is identical for all instances of that class/struct._____
___This array is often called the "vtable". The compiler tracks which function signature is_____
___placed at which offset into the array since all types derived from this type will invoke the_____
___offset in the array when accessing a virtual function with the same signature._____
___If a derived class overrides a virtual member function, then its vtable has a function pointer___
___to its overridden version instead of the base version. Since the same offset is called for the_____
___same signature code having a pointer/reference to a base type will always call the correct_____
___virtual function via its vtable._____

Q25) For a user-defined struct/class type `T`, write the member prototypes for the following: **[8 marks]**

Member	**Prototype**
Default constructor	T();
Copy constructor	T(T const&);
Copy assignment operator	T& operator =(T const&);
Move constructor	T(T&&);
Move assignment operator	T& operator =(T&&);
Destructor	~T();

Q26) Briefly explain how a programmer would (conceptually and) properly use the Resource Acquisition Is Initialization (RAII) design pattern. **[2 marks]**

He/she would ensure that the constructor allocated/acquired any needed resources and the destructor would deallocate/release any resources allocated/acquired during the lifetime of the object.

Part II: General Questions (5 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**.

NOTE: *Unless you are asked to write a full-and-complete program, assume* `using namespace std;` *is at the* ***top*** *of the code fragment you are writing. If you are writing a code fragment within a function, assuming the proper* `#include` *files have been included elsewhere. You may use C++11 or C++98 code in your answers unless otherwise prohibited.*

Q20) Write the full-and-complete program that would be placed inside `main()` (include any variable declarations needed) to read in from standard input an **unknown** number of `int`s and output to standard output their sum. You are not allowed to use any containers. You are only allowed to declare **at most two** variables and can only use **one loop** construct. Your program must work correct even in the presence of input failures and errors (i.e., stop summing on the first failure, error, or EOF). You do not need to worry about integer overflow. **[5 marks]**

```
#include <iostream>

int main()
{
  using namespace std;

  int sum = 0;
  for (int i; cin >> i; sum += i)
    ;

  cout << sum << '\n';

  return 0;
}
```

This page was intentionally left blank.

You may use it for rough work, or, if you've run out of space for a question.