03-60-340-30
2014 Winter, Mon. Feb. 10, 2014 in DH 355
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 **50 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 *The C++ Programming Language, 4th Edition*.
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 **6 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: 43

Copyright © 2014 Paul Preney. All Rights Reserved. No part of this document may be duplicated without permission.

Part I: Multiple Choice and Short Answer Questions (33 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) C++ was first standardized in _____. **[1 marks]**

(a) 1989 (b) 1998 (c) 1999 (d) 2003 (e) 2007 (f) 2011

Q4) The current C++ standard in effect was standardized in what year?

Answer: __2011_____ **[1 mark]**

Q5) True or False: A general aim of the C++'s language design is that it is more important to provide a useful feature than to prevent every misuse. **[1 mark]**

(a) True (b) False

Q6) True or False: C++ allows implicit violations of the type system. **[1 mark]**

(a) True (b) False

Q7) Briefly describe the zero overhead language design rule in C++. **[2 marks]**

__You don't pay for what you don't use._____

__If you don't use a language feature, you don't incur run-time costs for such features._____

_____

Q8) The imperative programming paradigm defines computation in terms of _____. **[2 marks]**

__programming statements that describe changes in state_____

_____

Q9) The modular programming paradigm allows _____: the ability to expose/hide functions/types defined in different modules.

Answer: __encapsulation_____ **[1 mark]**

Q10) The difference between the object-based and the object-oriented paradigms is that the object-based paradigm does not use/support _____.

Answer: __inheritance_____ **[1 mark]**

Copyright © 2014 Paul Preney. All Rights Reserved. No part of this document may be duplicated without permission.

Q11) Match the following descriptions with the programming paradigm that best applies.

Letter	Description
A	"Decide which classes you want; provide a full set of operations for each class; make commonality explicit using inheritance."
B	"Decide which procedures you want; use the best algorithms you can find."
C	"Decide which algorithms you want; parameterize them so that they work for a variety suitable types and data structures.
D	"Decide which modules you want; partition the program so that data is hidden within modules."
E	"Decide which types you want; provide a full set of operations for each type."

Write the letter of the most appropriate description corresponding to the programming paradigm below:

___E_____ Object-Based Paradigm **[1 mark]**

___C_____ Generic Paradigm **[1 mark]**

___D_____ Modular Paradigm **[1 mark]**

___A_____ Object-Oriented Paradigm **[1 mark]**

___B_____ Procedural Paradigm **[1 mark]**

Q12) True or False: C++'s template mechanism is Turing-complete. **[1 mark]**

(a) True (b) False

Q13) C++ is a multi-_____ programming language.

Answer: ___paradigm_____ **[1 mark]**

Q14) 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 an read-write instance of T._____

___T const* is a read-write pointer to a constant read-only instance of T._____

_____

_____

Copyright © 2014 Paul Preney. All Rights Reserved. No part of this document may be duplicated without permission.

Q15) Write a C++11 lambda function that accepts a `double` as an argument and returns its value multiplied by 3.14. **[2 marks]**

```
[](double d) { return d*3.14; }
```

Q16) The mathematician that designed key portions of the STL is _____ (full name).

Answer: _Alexander Stepanov_ **[1 mark]**

Q17) Briefly describe what a C++ Standard Library container represents. **[1 mark]**

A container is a collection of objects/values.

Q18) Briefly describe what a C++ Standard Library iterator represents. **[1 mark]**

An iterator in C++ represents a pointer to an object/value stored within a container, or,
it represents the one-past-the-end iterator position of the container. Iterators permit
through the elements/objects stored in the container.

Q19) C++ Standard Library's iterators were modeled upon which C language construct?

Answer: _pointers (and pointer arithmetic)_ **[1 mark]**

Q20) Briefly explain what is meant by a predicate in the C++ Standard Library. **[1 mark]**

A predicate is a function that returns a bool (true/false) value.

Q21) True or false: In C++ a Standard Library function requiring a predicate allows the predicate to be stateful. **[1 mark]**

(a) True (b) False

Q22) In the C++ Standard Library, all sorting operations rely on a _____ (3 words; Hint: SWO) to sort things. The default overloaded operator to perform such is _____.

3-word answer: _strict weak order_ **[1 mark]**

Operator answer: _less than_ **[1 mark]**

Copyright © 2014 Paul Preney. All Rights Reserved. No part of this document may be duplicated without permission.

Part II: General Questions (10 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.*

Q23) Your boss has just let go of an employee that wrote an insertion sort routine, called `insertion_sort()`, function using `std::vector` because he had asked for it to be written with `std::list`. Your boss thought he could just change the `typedef` from `vector<int>` to `list<int>` but got all kinds of compiler errors! He has **asked you** to **rewrite `insertion_sort()`** function to **use iterators** so that it **works properly for all bidirectional iterator inputs and random-access iterators**. He included a sample `main()` so you could see how it will be invoked. Pseudocode for the insertion sort is as follows, given an array A whose first index is zero:

```
for i from 1 to length(A) {
        j = i
        while (j != 0 and A[j-1] > A[j]) {
                swap(A[i], A[j-1])
                j = j – 1
        }
}
```

Your solution must only use valid bidirectional iterator operations. All comparisons must be done using < (i.e., less than). You must only use the for loop construct –not the while, do..while, or goto constructs.

NOTE: Write your answer on the next page. `insertion_sort()` will sort [first,last) in ascending order.

```
#include <iostream>
#include <list>
#include <algorithm>

using namespace std;

template <typename Iter>
void insertion_sort(Iter first, Iter last)
{
  // YOUR ANSWER GOES HERE.
}

int main()
{
  typedef list<int> LIST;
  LIST stuff{ 32, 14, 45, -24, 6543, 7635, 2, -5, -23, -25, -242 };
  insertion_sort(begin(stuff), end(stuff));
  for (auto const& i : stuff)
    cout << i << ' ';
  cout << endl;
}
```

[10 marks]

Copyright © 2014 Paul Preney. All Rights Reserved. No part of this document may be duplicated without permission.

Q23 Answer - Write **ONLY** the code **INSIDE** the provided **`insertion_sort()`** function above. (If you do otherwise then marks will be deducted.) You cannot change the function prototype.

```
// No need to sort if empty
if (first == last)
  return;

// No need to sort empty list...
auto second = next(first);
if (second == last)
  return;

// Above code is needed; i.e., some code to ensure that one does not go past
// the end when determining second.

for (auto i=second; i != last; ++i)
{
  for (auto j=i; j != first && *j < *prev(j); --j)
    iter_swap(j, prev(j));
}

// NOTE: Typo in pseudocode A[i] should be A[j]. This affects
//       iter_swap() code. A[i] results in iter_swap(i, prev(j));
```