03-60-340-30

2015 Winter, Tues. Feb. 10, 2015 in TC 204

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 **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 **9 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: 67

Part I: Multiple Choice and Short Answer Questions (38 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++ is a general-purpose, multi-paradigm dynamically-typed programming language. **[1 mark]**

(a) True [✓ (b) False]

Q2) A design objective of C++ is that its evolution must be driven by abstract problems. **[1 mark]**

(a) True [✓ (b) False]

Q3) A design objective of C++ is that it is more important to allow a useful feature than to prevent every misuse. **[1 mark]**

[✓ (a) True] (b) False

Q4) A design objective of C++ is to not allow any explicit violations of the static type system. **[1 mark]**

(a) True [✓ (b) False]

Q5) A design objective of C++ is to incur overhead for language features not used in a program. **[1 mark]**

(a) True [✓ (b) False]

Q6) Complete the sentence: The key difference between the object-based and object-oriented (OO) paradigm is the OO paradigm _____.

Answer: __supports inheritance (whereas object-based does not)_____________________ **[1 mark]**

Q7) The _____ programming paradigm uses encapsulation via translation units, namespaces, structs, unions, and/or classes in C++ but it is limited in that there can **only be one run-time** instance of it.

Answer: __modular____ **[1 mark]**

Q8) The _____ in C++ enables very powerful forms of compile-time polymorphism with the focus placed on algorithms and type abstractions as patterns.

Answer: __generic programming paradigm____ **[1 mark]**

Q9) The _____ programming paradigm defines computation in terms of programming statements that describe changes in state.

Answer: ___procedural____ **[1 mark]**

Q10) Write the C++ symbol name(s) that `stderr` is conceptually equivalent to.

Answer: ___std::cerr_ _std::clog____ **[1 mark]**

Q11) Write the C++ symbol name(s) that `stdout` is conceptually equivalent to.

Answer: __std::cout____ **[1 mark]**

Q12) Write the C++ symbol name(s) that `stdin` is conceptually equivalent to.

Answer: __std::cin____ **[1 mark]**

For Q13a through Q13f, your boss has asked you to transliterate a Java program into C++. Full full marks, ensure your answer uses correct C++ terminology and is appropriately detailed / specific.

Q13a) All inheritance in Java would be implemented as _____ inheritance in C++.

Answer: __virtual public____ **[1 mark]**

Q13b) All static methods in Java would be implemented as _____ member functions in C++.

Answer: __static____ **[1 mark]**

Q13c) All non-static methods in Java would be implemented as _____ member functions in C++.

Answer: __virtual____ **[1 mark]**

Q13d) All interfaces in Java would be implemented as _____ in C++.

Answer: __abstract classes____ **[1 mark]**

Q13e) All abstract classes in Java would be implemented as _____ using _____ _____ functions in C++.

First blank: __abstract classes____ **[1 mark]**

Second & third blanks: __pure______________________ __virtual___________________ **[1 mark]**

Q13f) Most enumerations in Java would be implemented as _____ in C++11. (Be specific!)

Answer: __enum class____ **[1 mark]**

Q14) Public inheritance should **only** be used to model _____ _____ relationships.

Answer: __is________________________________ __a___________________________ **[1 mark]**

Q15) In C++, the `this` symbol is always a _____ _____ to the type it represents.

Answer: __constant_________________________ __pointer_______________________ **[1 mark]**

Q16) In C++, by default all **struct inheritance** is _____ and all **members** are declared with _____ access.

Answer: __public____________________________ __public_______________________ **[1 mark]**

Q17) In C++, by default all **class inheritance** is _____ and all **members** are declared with _____ access.

Answer: __private___________________________ __private______________________ **[1 mark]**

Q18) The acronym's **RAII** expansion in words is: ______.

Answer: __Resource Acquisition Is Initialization_______________________________________ **[2 marks]**

Q19) Most C++ compilers implement exceptions in a way that the **cost** is _____ as long as exceptions are **not** thrown.

Answer: __essentially zero____ **[1 mark]**

Q20) Complete the sentence: The C++ language does **not** need the equivalent of Java's `finally` clause **because** ______. (Ensure your answer also justifies the statement.) **[3 marks]**

__C++ has destructors which (by convention) clean up object-held resources when the call_______

__stack is unwound____

____

____

Q21) Indicate whether or not each of the following **expressions** are **L-values** or **R-values** or an error (e.g., is illegal in the language). **[5 marks; 1 mark each]**

C++11 Expression	**Answer: L(value) or R(value) or E(rror)**
int(5)	R
double d = 3; (Answer with respect to d.)	L

float f = 2.3F;
float& r = f;
(Answer with respect to r.)
	L

long double a = -3.2e10L;
long double&& b = a;
(Answer with respect to b.)
	E

short&& m = 345; (Answer with respect to m.)	L

Q22) In Assignment 2, you manipulated an anonymous **union**:

```
class parse_value
{
private:
  parse_value_type type_;
  union
  {
    char charval_;
    std::string stringval_;
    long long integerval_;
    long double realval_;
  };
  // ...
};
```

Explain (i) what the differences are between a `union` and a `struct` and (ii) what that allowed you to accomplish in Assignment 2. **[4 marks]**

Answer (i): __A union's members all overlap in memory with each member's starting address_____

__being the same as all other members; struct members are non-overlapping wrt memory._______

__Only one member of a union can be stored at a time. All members of a struct are stored_________

__simultaneously.____ **[2 marks]**

Answer (ii): __Using a union allowed one to store zero or one **value** of a set of types___________

__i.e., char, string, long long, and long double, within any instance of parse_value._____________

__(This was accomplished without using pointers, inheritance, etc. using placement new_________

__instead.)____

____

____ **[2 marks]**

Part II: General Questions (29 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**.

Q23) Your boss has asked you to convert his C program:

```
#include <stdio.h>
#include <stdlib.h>

int value = 0;

int func(int* const p)
{
  *p += value;
  value = *p;
  return abs(*p * 2);
}

int main()
{
  int i;
  while (scanf(“%i”, &i) == 1)
    printf(“%i “, func(&i));
  printf(“\n”);
  return 0;
}
```

into a C++11 program replacing everything in C that has suitable C++ equivalents. Note that this means:

- you cannot use the C library at all in your answer,
- you need to replace everything that can be to a suitable C++ language construct –provided it is semantically correct to do so and it simplifies the code/syntax, and,
- everything else must remain as-is.

Your answer must be a **full and complete valid C++11 program. [10 marks]**

```
#include <iostream>
#include <cstdlib>

int value{};

int func(int& p)
{
  p += value;
  value = p;
  return std::abs(p * 2);
}

int main()
{
  int i;
  while (cin >> i)
    cout << func(i);
  cout << ‘\n’;
  return 0;
}
```

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

Q24a through Q24f make use of the following given struct:

```
struct X
{
  double *n;      // Must be dynamically allocated below
};
```

All answers must be written as if there were **written inside** X's definition. All dynamic allocation manipulations must be appropriately loaded, stored, or applied to the **member variable** n.

You may **assume** "using namespace std;" and any and all necessary #include files **appear elsewhere**. **Only** write the necessary code in your answer.

Q24a) Write the default constructor for X. It must **dynamically** allocate a **single double** variable set to **3.14**. Absolutely no code can appear between the constructor's braces (i.e., { and }). **[3 marks]**

```
X() :
  n(new double(3.14))
{
}
```

Q24b) Write the copy constructor for X. It must dynamically allocate a single double variable whose value is copied from the object passed to the constructor. Absolutely no code can appear between the constructor's braces (i.e., { and }). **[3 marks]**

```
X(X const& x) :
  n(x.n != nullptr ? new double(*x.n) : nullptr)
{
}
```

Q24c) Write the copy assignment operator for X with the correct semantics. It does not have to be exception-safe. **[4 marks]**

```
// NOTE: An exception safe version follows (other solutions possible)...
X& operator =(X const& x)
{
  if (this != &x)
  {
    X tmpX = x;    // Make copy

    // Swap tmpX.n with this->n...
    double *tmp = tmpX.n;
    tmpX.n = n;
    n = tmp;

    // and tmpX destructor will destroy its n (which is what was in X)
  }
  return *this;
}
```

Q24d) Write the move constructor for X with the correct semantics. **[3 marks]**

```
X(X&& x) :
  n(x.n)
{
  x.n = nullptr;
}
```

Q24e) Write the move assignment operator for X with the correct semantics. **[4 marks]**

```
X& operator =(X&& x)
{
  if (this != &x)
  {
    double *tmp = n;           // Save old n pointer value
    n = x.n;                   // Copy x.n into n
    x.n = nullptr;             // Null out x.n
    delete tmp;                // Free memory tied to tmp
  }
  return *this;
}
```

Q24f) Write the destructor for X with the correct semantics. **[2 marks]**

```
~X()
{
  delete n;
}
```