600.120: Intermediate Programming
Midterm 2

Peter H. Fröhlich
phf@cs.jhu.edu

November 15, 2006
Time: 40 Minutes

Start here: Please fill in the following important information using a **permanent pen** before you do **anything** else! Your exam will **not** be graded if you use a pencil or erasable ink on this page.

Name (print): _____

Login (print): _____

Ethics Pledge: With your signature you **certify** the information above and you also **affirm** the following: "*I agree to complete this exam without unauthorized assistance from any person, materials, or device.*"

Signature: _____

Date: _____

Instructions: Please read these instructions carefully before you start. **Switch off** your phones, pagers, and other noisy gadgets! You are **not** allowed to have anything but a pen (pencil, eraser) and this exam on your desk. You are **not** allowed to talk to anyone during the exam. If you have a question, please raise your hand **quietly**. You must **remain seated quietly** until all exams have been collected. Remember that you can **not** claim grading errors if you do not use a **permanent** pen for your answers.

Do not open before you are told to do so!

You got _____ out of 40 points.

1 Binary Warmup (12 points)

For each of the following statements, determine whether it is either **true** or **false**. **(1 point each)**

1. Abstract classes in C++ must have at least one pure virtual member function.

2. Assume `int i=42`; the expression `0 < i < 20` evaluates to `true` (or `1` if you prefer).

3. Exceptions are particularly useful to indicate that a constructor has failed.

4. The new C++ operators for casting are generally faster than the old ones.

5. A function can be declared inside another function.

6. In C++ classes cannot be nested (i.e. "classes within classes").

7. One purpose of exceptions in C++ is to simplify the structure of error-handling code.

8. The C language is based on the B language which in turn was based on the BCPL language.

9. Templates in C++ are resolved and checked at compile-time and link-time.

10. In C, the `->` operator is defined by the following equivalence: `x->y` ⇔ `(*x).y`.

11. The purpose of UML class diagrams is to show as many implementation details as possible.

12. The new C++ operators for casting are easier to find in source code than the old ones.

2 Tough Choices (8 points)

For each of the following questions, circle **one** answer out of the choices given. **(2 points each)**

1. How can you keep C++ from using certain constructors to perform automatic conversions?
   (a) You have to make those constructors `private`.
   (b) You have to make those constructors `explicit`.
   (c) You have to make those constructors `protected`.
   (d) You have to make those constructors `static`.
   (e) None of the above.

2. Which of the following is **not** a valid storage class for **variables** in C?
   (a) `register`
   (b) `static`
   (c) `extern`
   (d) `auto`
   (e) None of the above.

3. How does Subversion help a team of developers collaborate?
   (a) Subversion keeps track of previous revisions of all files, providing a huge "undo" button.
   (b) Subversion keeps track of the changes made and notifies developers of conflicting changes.
   (c) Subversion stores a history of changes and allows developers to review the progress made.
   (d) All of the above.
   (e) None of the above.

4. Assume `int a[10]` and `int b[10]`; what does the instruction `a = b` do in this context?
   (a) All values stored in the array `b` are copied into the array `a`.
   (b) A pointer to the first element of the array `b` is copied into the pointer `a`.
   (c) The first element of the array `b` is copied into the first element of the array `a`.
   (d) The instruction is not legal.
   (e) None of the above.

3 Short Answer (8 points)

For each of the following questions, answer in **one to three** sentences, the shorter the better. **(2 points each)**

1. C++ allows for **multiple inheritance** which means that a subclass can have more than one superclasses. In this context, the mechanism of **virtual inheritance** becomes important. Give an example of multiple inheritance where virtual inheritance is essential. **Explain!**

2. Congratulations, you found a job! As you make your way through thousands of lines of code, you run into the obscure instruction `i = i[a]`. You find that these identifiers are declared as `int a[10]` and `int i`. The compiler spits out no warnings, none of the test cases fail, and coverage analysis shows that the line actually runs. What's going on? **Discuss!**

3. This is a **two-part** question. First, under what circumstances is it **necessary** to define copy constructors and assignment operators? Second, what is the **difference** between copy constructors and assignment operators? **Explain!**

4. The keyword `const` in C++ is used for a lot of different things. Give an example that shows at least **three** distinct uses of `const` and discuss what they mean. **Explain!**

4 Generic Reversals (6 points)

Give a C++ function `reverse` that takes a generic `std::vector<T>` and **reverses** the vector **in place**. In other words, given an empty `std::vector<int> a` into which the values 10, 20, 30, and 40 have been `push_back()`ed, after the call `reverse(a)` the elements of `a` are now 40, 30, 20, and 10 respectively.

5 Objective Examination (6 points)

Draw a UML **class diagram** for the following problem domain: *"An exam consists of a number of questions. There are four kinds of questions: true/false, multiple choice, short answer, and mystery. Each question is worth a number of points."* Keep it **simple**: Focus on classes, attributes, generalizations, and associations. Please **document** any further assumptions you make.

This page is intentionally **mostly** blank in case you run out of space elsewhere. If you ended up here early, please go over **everything** again and remain seated **quietly**! Make sure that the title page is filled out correctly and in **permanent** pen. Maybe you want to "rewrite" your **answers** in permanent pen as well?