



**Midterm Examination  
Winter TERM 2009**

**Computer Science 246  
Software Abstraction and Specification  
Sections 01**

**Duration of Exam: 110 minutes  
Number of Exam Pages (including cover sheet): 3  
Total number of questions: 8  
Total marks available: 72**

**CLOSED BOOK, NO ADDITIONAL MATERIAL ALLOWED**

**Instructor: Ric Holt  
February 26, 2009**

1. (a) **1 mark** True or False : After executing the following code in a function, the output produced is necessarily 0 (zero).

```
int j;
cout << j;
```

- (b) **1 mark** True or False : The following code causes 0 (zero) to be printed.

```
int x = 0;
if ( x = 0 )
    cout << x;
else
    cout << 10;
```

- (c) **3 marks** Re-write the following code so it uses a **while** loop (not a **do-while** loop) instead of a **for** loop.

```
int i;
for ( i = 0; i < LIMIT; i += 1 )
    cout << i << endl;
```

- (d) **3 marks** Write a **switch** statement that prints the string "zero" when variable i is 0, the string "five" when i is 5, and otherwise prints the string "No go".

2. (a) **1 mark** True or False : The following code outputs the numbers 0, 2 and 4 (along with some blanks).

```
for ( int j = 0; j < 3; j += 1 )
    cout << 2 * j << " ";
```

- (b) **3 marks** Give code that generates and prints a random integer whose values is 2, 3 or 4. Assume your code is inserted into the main function. Assume any needed **#include** directives are provided.

- (c) **3 marks** What is a *precondition*? What is it used for?

- (d) **3 marks** Describe *call by value*.

3. (a) **3 marks** What is an *assert* and what is it used for? Give an example.

- (b) **3 marks** Below is a function named `ave`. Give an overloaded version of `ave` that accepts 3 doubles and returns their average.

```
double ave( double n1, double n2 )
{ return (n1 + n2) / 2.0; }
```

- (c) **1 mark** True or False : The following declaration creates an array with 4 elements.

```
double d[4];
```

- (d) **3 marks** Write a function that returns the total of the values in an array of doubles. Handle the array size as a parameter in a flexible way.

4. (a) **1 mark** True or False : A mutator changes a variable from being public to private.

- (b) **3 marks** What does *ADT* stand for? What are the key aspects of an ADT?

- (c) **3 marks** Compare and contrast arrays versus vectors in C++.

- (d) **5 marks** Write a **complete** C++ program that does the following: reads in a sequence of 11 names, which are treated as C++ strings (not C-strings) and prints them out, one per line, in reverse order.

5. (a) **3 marks** What is a *dependent object*? If you have dependent objects in your program, what three things should you explicitly implement? Hint: one of these things is assignment operator “=” overloading.
- (b) **3 marks** How does the compiler recognize that a constructor is a copy constructor? When does the compiler automatically call a copy constructor?
- (c) **3 marks** You are to provide declarations along with allocating (**new**) code so that `M[i][j]` indexes matrix `M`. The size of `M` has already been read into a variable named `N` at runtime. Note the matrix is a square (`N` by `N`). You do not need to put values into the matrix.
6. (a) **3 marks** List three things that you expect to find in an interface (.h) file.
- (b) **3 marks** Explain with an example how `#ifndef` is used to deal with multiple indirect inclusion of interface files.
- (c) **3 marks** What is an *object module*?
- (d) **3 marks** With simple examples, explain the three ways of accessing items in **namespaces**.
7. (a) **2 marks** Compare and contrast the *global name space* and the *nameless name space*.
- (b) **3 marks** Give a simple but **complete** C++ program that separately outputs the strings "Hello" and then "Bob" separated by a new line character, to a text file named “robert.txt”. Use a stream object to do this.
8. **6 marks** Consider a class called `polynomial` (as in Assignment 3) in which a C++ vector named `v` is used to represent a polynomial. Give the implementation of this class’s multiply function (which overloads `*`) for this polynomial class.

```
// Prototype for the multiply function
polynomial polynomial::operator *(const polynomial &rtSide)
// Give comments to describe any parts you cannot complete in C++
```