

University of Toronto  
Faculty of Applied Science and Engineering

ECE 244F

PROGRAMMING FUNDAMENTALS

Fall 2012

Midterm Test

Examiners: T.S. Abdelrahman, V. Betz, and M. Stumm

Duration: 110 minutes

This test is OPEN Textbook and CLOSED notes. The use of computing and/or communicating devices is NOT permitted.

Do not remove any sheets from this test book. Answer all questions in the space provided. No additional sheets are permitted.

Work independently. The value of each question is indicated. The total value of all questions is 100.

Write your name and student number in the space below. Do the same on the top of each sheet of this exam book.

Name: \_\_\_\_\_  
(Underline last name)

Student Number: \_\_\_\_\_

- |           |            |
|-----------|------------|
| Q1. _____ | Q7. _____  |
| Q2. _____ | Q8. _____  |
| Q3. _____ | Q9. _____  |
| Q4. _____ | Q10. _____ |
| Q5. _____ | Q11. _____ |
| Q6. _____ | Q12. _____ |

Total

**Question 1. (9 marks).** *General. Prepared by Professors Abdelrahman and Betz.*

Answer the following questions by circling either **Yes** or **No**, or by providing a **very brief and direct** answer when indicated. There is no need to justify your answer.

- (a) What is the name of the Integrated Development Environment (IDE) software tool you use in the lab to compile and run your programs?

Write the name here:

- (b) Every class has at least one constructor function, even when none is declared.

Circle one answer:    Yes            No

- (c) What is the default access level assigned to members of a class, `private` or `public`?

Circle one answer:    `private`    `public`

- (d) A default version of the equality test operator, `operator==`, is created automatically for a C++ class if the class does not define one.

Circle one answer:    Yes            No

- (e) Memory allocated with the `new` operator using `int *p = new int;` should be deallocated by calling the `delete[]` operator when it the memory is no longer needed.

Circle one answer:    Yes            No

- (f) Friend functions can access the private data members of a class, but are not member functions of that class.

Circle one answer:    Yes            No

- (g) Every C++ program must contain exactly one global function named `main`.

Circle one answer:    Yes            No

- (h) `ifstream inFile ("fname.txt");`  
The line of code above will open the file `/fname.txt`; that is, a file in the root directory of the file system named `fname.txt`.

Circle one answer:    Yes            No

- (i) Stream variables (e.g. types `istream`, `ostream`, and `sstream`) must be passed by reference in function calls.

Circle one answer:    Yes            No

**Question 2. (4 marks).** *Pointers. Prepared by Professor Abdelrahman.*

What is the output of the following code?

```
int* p;  
int* q;  
p = new int;  
*p = 43;  
q = p;  
*q = 52;  
p = new int;  
*p = 78;  
q = new int;  
*q = *p;  
cout << *p << " " << *q << endl;
```

Write the output below:

**Question 3. (10 marks).** *Classes. Prepared by Professor Abdelrahman.*

(a) (5 marks). Consider the following definition:

```
class xClass {
    private:
        int u;
        double w;

    public:
        void func();
        void print() const;
        xClass();
        xClass(int a, double b);
};
```

(i) How many members does `xClass` have?

(ii) How many private members does `xClass` have?

(iii) How many constructors does `xClass` have?

(iv) Write a C++ declaration that declares a variable `t` of type `xClass` and initializes its data members to 20 and 40.5 respectively.

(b) (5 marks). Find the syntax errors in the definitions of the following classes. Indicate the error by placing an “X” to the left of the line that has the error. If there are no errors, write “No Errors” next to the class name.

(i) 

```
class AA {
    private:
        int x;
        int y;
    public:
        void print() const;
        int sum();
        AA();
        int AA(int a, int b);
};
```

```
(ii) class BB {
    int one;
    int two;
    public:
        bool equal() const;
        print();
        BB();
};
```

```
(iii) class CC {
    private:
        int u;
        int v;
    public:
        void set(int a, int b);
        void print() const;
        CC();
        CC(CC source);
};
```

**Question 4. (6 marks).** *Compilation. Prepared by Professor Betz.*

A program consists of 3 .cpp files, and 2 .h files, as listed below. All the files are in the same directory.

```
X.h
#include "Y.h"
class X {
private:
    Y y1;
public:
    X();
};
```

```
Y.h
class Y {
private:
    int i1;
public:
    Y();
};
```

```
X.cpp
#include "X.h"
X::X () {
}
```

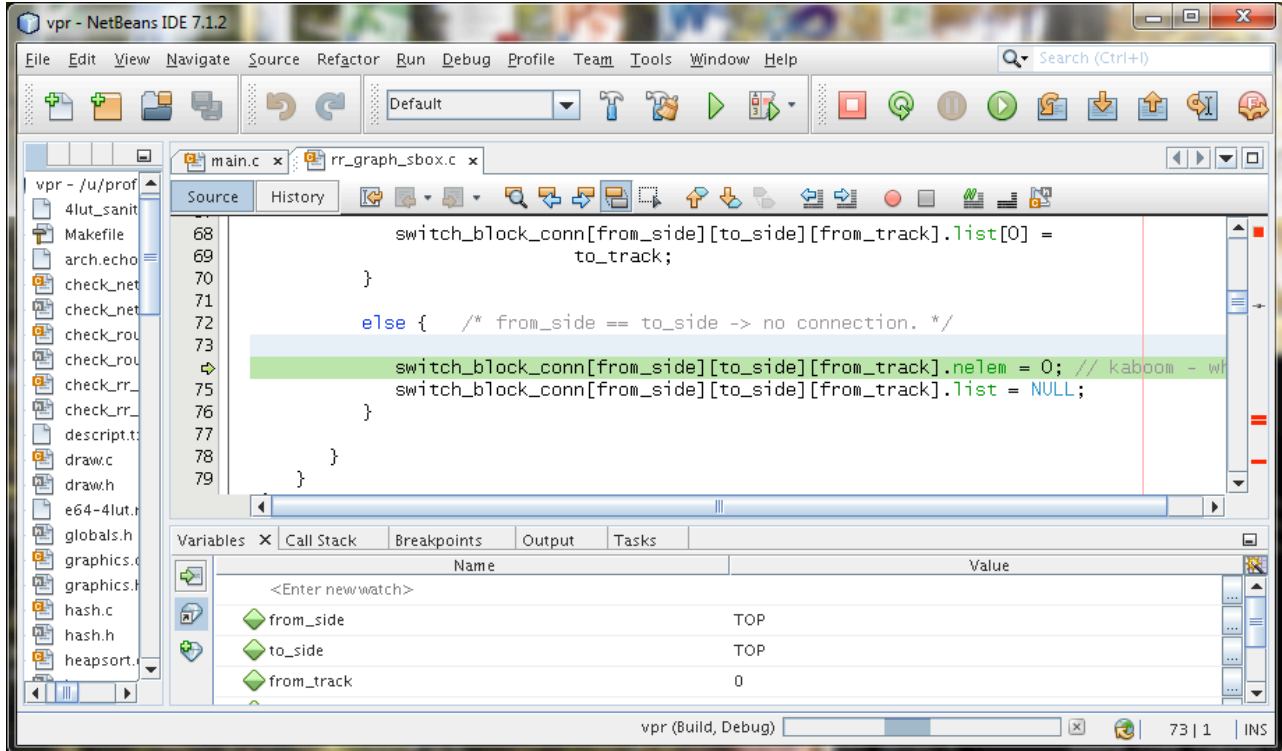
```
Y.cpp
#include "Y.h"
Y::Y () {
}
```

```
main.cpp
#include "X.h"
#include "Y.h"

int main () {
    X x2;
    Y y2;
    return (0);
}
```

- (a) (2 marks). If you attempt to compile the files above, a compiler error will be generated. What is the cause of the error? Describe the best way to modify the code to fix the error.
- (b) (2 marks). Write the command line to compile this program from the source files above, and produce an output file called *main.exe*. Recall that the name of the compiler we are using is `g++`, and that the `-o <outputFileName>` option is used to choose a name for the output executable.
- (c) (2 marks). Assume that prior commands have created *X.o*, *Y.o* and *main.o*. Now a change is made *only* to *main.cpp*. Write out a sequence of two commands that can be used to recompile only the necessary .cpp file(s), and to link all the object files together to create *main.exe*. Recall that the `-c` option instructs `g++` to compile a source file into an object file and then stop without linking the object files together into the executable program.

**Question 5. (6 marks).** *Debugging. Prepared by Professor Betz.*



You wish to use the NetBeans debugger to debug a program, which has a segmentation fault when it is run.

- (2 marks). Assuming NetBeans is running and the project is open; describe how to start the debugger. You may either circle and label by “A” the appropriate icon in the picture above, **or** describe below the menu command(s) used to start the debugger.
- (2 marks). The program experiences a segmentation fault while running in the debugger, and you select “Discard and Pause” so you can examine the program state. Describe how to determine the sequence of subroutines that have been called to reach the line at which the program seg faults. You can either describe below the appropriate menu command(s), **or** circle and label with “B” the appropriate icon in the picture above and describe its use.
- (2 marks). Describe below in one sentence what a breakpoint does.

**Question 6. (9 marks).** C++ I/O. Prepared by Professor Abdelrahman.

The following program reads 10 integers from the standard input until either an invalid integer is given or the end-of-file is reached. The 10 integers appear on a single input line. Re-write the program so it used a `stringstream` for input rather than `cin`.

```
#include <iostream>
using namespace std;
int main() {
    int i;
    int value;
    for (i=0; i < 10; ++i) {
        cin >> value;
        if (cin.fail()) {
            if (cin.eof()) {
                cout << "More integers expected." << endl;
                return (0);
            }
            else {
                cout << "Invalid integer." << endl;
                return (0);
            }
        }
        cout << "Integer value is: " << value << endl;
    }
    return (0);
}
```

Write your code in the box below.



**Question 7. (6 marks).** *Constructors and Destructors. Prepared by Professor Betz.*

```
#include <iostream>
using namespace std;

class E {
public:
    E() {cout << "E created\n"; }
    E(const E& src) {cout << "E copy created\n"; }
    ~E() {cout << "E destroyed\n"; }
};

E e1;
E e2 = e1;

int main () {
    cout << "Starting main\n";
    E e3[2];
    cout << "Ending main\n";
}
```

Write out the output generated by the program listed above.

**Question 8. (10 marks).** *Constructors and Destructors. Prepared by Professors Abdelrahman and Betz.*

Consider the following definition and implementation of a class that represents a day of the year.

```
using namespace std;
#include <iostream>

class DayOfYear {
private:
    int day;
    int month;

public:
    ~DayOfYear(); // method 1
    DayOfYear(int d, int m); // method 2
    DayOfYear(const DayOfYear & other); // method 3
    int getDay(); // method 4
    int getMonth(); // method 5
    void setDay(int d); // method 6
    void setMonth(int m); // method 7
    DayOfYear & operator=(const DayOfYear & src); // method 8
    void print(); // method 9
};

DayOfYear::~DayOfYear(){} // Nothing to do

DayOfYear::DayOfYear(int d, int m) {
    day = d;
    month = m;
}

DayOfYear::DayOfYear(const DayOfYear & other) {
    day = other.day;
    month = other.month;
}

int DayOfYear::getDay() {return (day);}

int DayOfYear::getMonth() {return (month);}

void DayOfYear::setDay(int d) {day = d;}

void DayOfYear::setMonth(int m) {month = m;}

DayOfYear & DayOfYear::operator=(const DayOfYear & src) {
    day = src.day;
    month = src.month;
    return(*this);
}

void DayOfYear::print() {cout << day << "/" << month << endl;}
```

Now consider the following code in the functions `main` and `flip`. The code is shown in a table to facilitate the writing of your answers. Indicate which method of the class is invoked when each line of the code is executed. There may be multiple methods invoked on a single line; indicate them all in the order in which they are invoked.

The methods are numbered in the class definition above, so just write the method numbers. If no methods are invoked, simply write **NONE**. There are no compile-time errors in the code.

Code	Class Methods Invoked
<code>using namespace std;</code>	NONE
<code>#include &lt;iostream&gt;</code>	NONE
<code>DayOfYear* flip(DayOfYear source);</code>	NONE
<code>int main() {</code>	NONE
<code>DayOfYear July8(8,7);</code>	
<code>DayOfYear August7(0,0);</code>	
<code>bool isMayOK = true;</code>	
<code>if (isMayOK) {</code>	
<code>DayOfYear* May19;</code>	
<code>May19 = new DayOfYear(19,5);</code>	
<code>DayOfYear* MothersDay = May19;</code>	
<code>delete May19;</code>	
<code>}</code>	
<code>August7 = *(flip(July8));</code>	
<code>return(0);</code>	
<code>}</code>	
<code>DayOfYear* flip(DayOfYear source) {</code>	
<code>DayOfYear* temp = new DayOfYear(0,0);</code>	
<code>temp-&gt;setMonth(source.getDay());</code>	
<code>temp-&gt;setDay(source.getMonth());</code>	
<code>return(temp);</code>	
<code>}</code>	

**Question 9. (8 marks).** *Class Definition. Prepared by Professor Abdelrahman.*

Consider the following definition of the class `Complex` and its use in the `main` function:

```
class Complex {
    private:
        float real;
        float imag;
    public:
        int getReal();
        int getImag();
        void setReal(float r);
        void setImag(float i);
        void print();
};

int main() {
    Complex a;

    a.setReal(1.0);
    a.setImag(-9.6);

    Complex b = a;
    Complex c;

    c = a;
    return(0);
}
```

A programmer decides to change the class definition into the following:

```
struct _complex {float real; float imag;};

class Complex {
    private:
        struct _complex* number;
    public:
        // May need to add new function members or delete some
        int getReal();
        int getImag();
        void setReal(float r);
        void setImag(float i);
        void print();
};
```

- (a) (2 marks). What additional (i.e., new) class member functions **must** be defined for the class to remain correct? Write only the prototypes of these functions. If no new members must be added, write **NONE**.
- (b) (2 marks). What existing member functions **must** be removed for the class to remain correct? Write only the prototypes of these functions. If no members must be removed, write **NONE**.
- (c) (2 marks). Which of the functions that were not removed (in part (b)) from the original class definition **must** be re-implemented to reflect the new definition of the class? Again, write only the prototypes of these functions. If none must be re-implemented, write **NONE**.
- (d) (2 marks). Identify the lines in `main` **must** be changed for `main` to continue to work correctly with the changes made to the class `Complex`. For each line that must change, write down the original line and what it should be changed into. If no lines must change, write down **NONE**.

**Question 10. (10 marks).** *Operator Overloading. Prepared by Professor Abdelrahman.*

The following class is used to create objects that represent ordinary fractions “n/d”, consisting of a numerator “n” and a denominator “d”.

```
using namespace std;
#include <iostream>

class Fraction {
private:
    int numerator;
    int denominator;

public:
    Fraction(int num, int denm);
    int getNumerator();
    int getDenominator();
    void setNumerator(int num);
    void setDenominator(int denm);
    void print();
};

Fraction::Fraction(int num, int denm) {
    numerator = num;
    // Should check that denm is not 0, but ignore for now
    denominator = denm;
}

int Fraction::getNumerator() {
    return (numerator);
}

int Fraction::getDenominator() {
    return (denominator);
}

void Fraction::setNumerator(int num) {
    numerator = num;
}

void Fraction::setDenominator(int denm) {
    // Should check that denm is not 0, but ignore for now
    denominator = denm;
}

void Fraction::print() {
    cout << numerator << "/" << denominator << endl;
}
```

We wish to overload the “+” and “\*” operators for the Fraction class to be able to write code like this in a non-member function (say main):

```
Fraction X(1,2);  
Fraction Y(2,3);  
:  
.. = X + Y;  
.. = X * Y;
```

For example, if X represents “1/5” and Y represents “4/6” then X + Y results in an object that represents “26/30” and X \* Y results in an object that represents “4/30”.

Write the implementation of the two overloaded operator functions as members of the class Fraction. Clearly show the function header and its body.

(5 marks). Write the overloaded + operator function here

(5 marks). Write the overloaded \* operator function here

**Question 11. (10 marks).** *Memory Management in Classes. Prepared by Professor Betz.*

Consider the following definition of the class IArray. This class is used to store and manipulate an array of integers.

```
class IArray {
private:
    int nElem;
    int *array;
public:
    IArray();
    IArray(int _nElem);
    void setEntry (int index, int value);
    int getEntry (int index);
    ~IArray();
    IArray& operator=(const IArray& rhs);
    // Other member functions not shown
};

IArray::IArray() {
    nElem = 1;
    array = new int[nElem];
    array[0] = 0;
}

IArray::IArray(int _nElem) {
    nElem = _nElem;
    array = new int[nElem];
    for (int i = 0; i < nElem; i++)
        array[i] = 0;
}

void IArray::setEntry (int index, int value) {
    array[index] = value;
}

int IArray::getEntry (int index) {
    return (array[index]);
}
```



(a) (2 marks). Write the implementation of the destructor for this class. Clearly show the function header and the body.

(b) (8 marks). Write the implementation for the (deep copy) assignment operator for this class. Clearly show the function header and the body.

**Question 12. (12 marks).** *Arrays and Objects. Prepared by Professor Abdelrahman.*

Consider the following definition of the class `Complex`:

```
class Complex {
    private:
        float real;
        float imag;

    public:
        Complex(); // Default constructor
        Complex(float r, float imag); // Second constructor
        int getReal();
        int getImag();
        void setReal(float r);
        void setImag(float i);
        void print();
};
```

- (a) (1 mark). Write a C++ declaration to declare an array called `A` of 20 `Complex` objects.
- (b) (1 mark). Which constructor is used to initialize each of the array elements in the declaration above? Write the name of the constructor (see class definition above), or NONE if no constructor is invoked.
- (c) (3 marks). Given an integer variable `n`, write C++ code to dynamically allocate an array of `n` `Complex` objects called `dA`.
- (d) (1 mark). Which constructor is used to initialize each of the array elements in the declaration in part (c) above? Write the name of the constructor (see class definition above), or NONE if no constructor is invoked.

- (e) (2 marks). Given an integer variable `n`, write C++ code to dynamically allocate an array of `n` pointers to `Complex` objects. Call the array `dpA`.
- (f) (4 marks). Write C++ code to dynamically allocate `n` `Complex` objects and have each object pointed to by an element of the array declared in part (e) above. Objects pointed to by even-indexed elements should be initialized using the default constructor. Objects pointed to by odd-indexed elements should be initialized using the second constructor with the real and imaginary parts set to the index value.