University of Toronto

Faculty of Applied Science and Engineering

Midterm – October, 2008

ECE244 --- Programming Fundamentals

Examiners: Courtney Gibson, Ashvin Goel, and Michael Stumm

Instructions: Please read carefully --- marks are deducted if not followed.

- Write you last name, first name, and student number in the fields below
- Write your name and student number at the top of every page of this exam.
- This exam has eleven (11) pages.
- No additional sheets are permitted.
- Do not remove any sheets from this exam.
- There are a total of 9 questions and the weight of each question is the same.
- All questions must be answered on these sheets.
- The use of calculators or computers is not permitted.

Last Name: _____

First Name: _____

Student Number: _____

Do not write below this line:

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*

1. Warmup questions

a. Consider the following code:

```
class Data {
  private:
     int a;
  public:
     int get() { return a; }
};

int main() {
     Data data;
     return 0;
}
```

Modify the code above so that `data` is declared as a global variable.

b. Identify what is wrong with the following code and fix the error:

```
class C {
      public:
         int x;
};

void main(void) {
   int x = 100;
   C *c;

   c->x = x+1;

   cout << c->x << endl ;
}
```

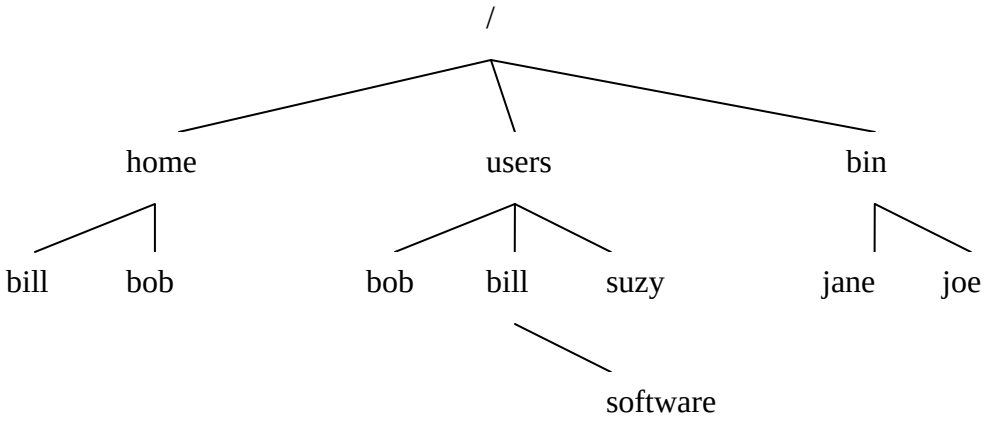
c. Identify what is wrong with the following code and fix the error:

```
const int MAX=10;

int list[MAX];

void main(void) {
   int sum = 0;

   for (int i = 0; i <= MAX; i++) {
      list[i] = i;
   }
   for (int i = 0; i <= MAX; i++) {
      sum  += list[i];
   }
   cout << sum << endl ;
}
```

2. Unix Directory Structure

Given the following directory hierarchy, assume the current working directory is /users/suzy. What complete Unix command would you use to switch the current working directory to /users/bill/software using as few characters as possible.

Answer:



3. Debugging

Please ***briefly*** describe (one sentence) what the "Step" command does in either gdb or DDD when debugging a running program:

Answer:

4. Makefiles

Consider a makefile with the following contents:

```
project: data.o driver.o io.o
        g++ –o project data.o driver.o io.o
data.o: data.cpp data.h
        g++ –c data.cpp
driver.o: driver.cpp data.h
        g++ –c driver.cpp
io.o: io.cpp
        g++ –c io.cpp
```

and the following listing of the directory with this makefile:

```
-rw-r--r-- 1 plab courses 673 Feb 4 19:43 Makefile
-rw-r--r-- 1 plab courses 673 Feb 4 18:10 data.cpp
-rw-r--r-- 1 plab courses 673 Feb 4 16:59 data.h
-rw-r--r-- 1 plab courses 673 Feb 4 17:00 data.o
-rw-r--r-- 1 plab courses 673 Feb 4 17:30 driver.cpp
-rw-r--r-- 1 plab courses 673 Feb 4 17:31 driver.o
-rw-r--r-- 1 plab courses 673 Feb 4 17:41 io.cpp
-rw-r--r-- 1 plab courses 673 Feb 4 17:40 io.o
-rwxr-xr-x 1 plab courses 673 Feb 4 18:01 project
```

Note that the date and time the file was last modified is listed just before the file name.

a. If your current working directory is the above listed directory and you issue the command “make”, which files will be modified (assuming that all compilations succeed)?

Answer:

b. Draw the “make” dependency diagram:

5. Memory Allocation

Consider the following definition of the Node class:

```
class Node {
  public:
    char *name;
    Node *left;
    Node *right;
    Node() { name = left = right = NULL; }
    Node(char *nm);
    ~Node();
};

Node::Node(char *nm){
    name = new char[strlen(nm) + 1];
    strcpy(name, nm);
    left = right = NULL;
}

Node::~Node(){
    delete [] name;
    delete left;
    delete right;
}
```

The following pieces of code use the Node class and may have one or more memory allocation bugs. These bugs can be of the following types: 1) double deletion, 2) memory leak, or 3) deletion of stack memory. Indicate a memory allocation bug, if it exists, by placing an X on the right side of the appropriate line of code where the bug is triggered ***in main()*** and putting a number 1, 2 or 3 depending on the type of bug. Assume that the Node class is defined correctly above.

(a)

```
int main() {
    Node *n = new Node;
    n->name = “first”;
    delete n;
}
```

(b)

```
int main() {
    Node *n = new Node(“first”);
    Node *m = new Node(”second”);
    *m = *n;
    delete n;
    delete m;
}
```

(c)

```
int
main() {
    Node *n = new Node(“first”);
    n->left = n->right = new Node;
    delete n;
}
```

6. Objects.

Study the following class definition and implementation.

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

struct _two {
     int a;
     int b;
};

class Two {
   private:
     struct _two *data;
   public:
     Two(int f = 0, int s = 0);
     Two(Two & rhs);
     Two & operator+=(Two rhs);
     void print();
};

Two::Two(int f, int s) {
     data = new struct _two;
     data->a = f;
     data->b = s;
}

Two::Two(Two& rhs) {
     data = new struct _two;
     data->a = rhs.data->a;
     data->b = rhs.data->b;
     rhs.data->a = data->b;
     rhs.data->b = data->a;
};

Two& Two::operator+=(Two rhs) {
     data->a += rhs.data->a;
     data->b += rhs.data->b;
     return *this;
}

void Two::print() {
     cout << "(" << data->a << "," << data->b << ")" << endl;
}
```

```
// Now consider the following main function, which uses the
//  above class.

int main() {
     Two a(1, 1);
     Two b(4, 16);
     Two c = b;
     Two d;

     a.print();      // Statement # 1
     b.print();      // Statement # 2
     c.print();      // Statement # 3
     d.print();      // Statement # 4
     a += b;
     d += c;
     a.print();      // Statement # 5
     b.print();      // Statement # 6
     c.print();      // Statement # 7
     d.print();      // Statement # 8
     return (0);
}
```

Indicate what is being printed by each statement in the above main function. For simplicity, each statement that produces output has been given a number, and you can write the output of each statement in the table below.

Statement #	Output
1	
2	
3	
4	
5	
6	
7	
8	

7. Variable Scope

What is the output of the following program. Please provide answer on the next page.

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

class X {
  private:
    int x ;
    int y ;
  public:
    X() { x = 0 ; y = 0 ; }
    X( int x, int y ) { this->x = x; this->y = y ; }
    static int z ;
    int get_x() { return x ; }
    int get_y() { return y ; }
    int foo( int&, int ) ;
} ;
int X::z = 1 ;

int X::foo( int& a, int b ) {
    int *y = new int ;
    *y = x ;

    cout << "foo starting " << a << " " << b << " " << x << endl ;
    x += 1 ;
    *y = a + b ;
    {
      int x = a + *y ;
      b = 34 ;
      {
        int y ;
        y = b + 2 ;
        a = x + y + this->x ;
      }
    }
    cout << "foo ending " << a << " " << b << " " << x << endl ;
    delete y ;
    return( x + b ) ;
}

main() {
    int a = 2 ;
    int b = 3 ;
    X x ;
    X y(2,3) ;

    cout << "1. " << a << " " << b << " " << x.get_x() << endl ;
    b = x.foo(a,3) ;
    cout << "2. " << a << " " << b << " " << x.get_x() << " "
                  << y.get_x() << endl ;
    b = y.foo(a,3) ;
    cout << "3. " << a << " " << b << " " << y.get_x() << " "
                  <<  y.get_y() << endl ;
}
```

8. I/O

Write a C++ function that repeatedly reads floating point numbers from the standard input (using cin) and then immediately outputs the input numbers (using cout), one number per line. The function is to return either: (1) when a non-floating point number is input; or, (2) when the end of file is reached. In the former case, “Invalid input” should be output on a line by itself, and in the later case “End of File reached” should be output on a line by itself before the function returns.

9. Classes with Pointers

The following class is used to keep track of songs in a music playlist:

```
class Song {
      private:
            int         track_number;
            double *    song;
            int         song_length;

      public:
            Song(int, int);
            ~Song() { delete song; }
};

Song::Song(int track, int length) {
    track_number = track;
    song_length = length;
    song = new double[song_length];
 }
```

a) Assume that the code above compiles correctly and that a and b are Song. ***Briefly*** explain (one sentence) why with the default assignment operator the assignment "`a = b;`" does not result in two independent copies of Song objects.

b) Please write a complete Song::operator= function that copies Song objects correctly.