University of Toronto

Faculty of Applied Science and Engineering

Midterm – November, 2010

ECE244 --- Programming Fundamentals

Examiners: Tarek Abdelrahman, Michael Gentili, 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 thirteen (13) 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*

Name: ____________________________________ Student Number________________

1. Warmup questions

Answer the following questions either by circling**Yes** or **No**, or by providing a **very brief** and direct answer when indicated.

a. **Yes** or **No**? If a program has passed all inputs you test it with, it has no bugs.

b. **Yes** or **No**? If a program has a bug, it will always show when running the program through the debugger.

c. **Yes** or **No**? The statement: `delete p`; de-allocates the pointer `p`.

d. **Yes** or **No**? The exercise command determines if your program is correct or not by comparing your source files to reference source files. If the files do not match, it issues an error message.

e. **Yes** or **No**? If the programmer does not write his or her own copy constructor, the compiler's copy constructor will be invoked during a call-by-value?

f. What is the UNIX command needed to make a directory `mydir` accessible only to the owner of the directory?

g. Give a name of your home directory on ECF.

h. What is the name of the software tool you use in the lab to convert your C++ code into an executable program?

i. Place an **X** to the left of each statement causing a compiler error and to the right say why

```
//in Circle.h
class Circle {
  public:
    Circle( Circle c ) ;
    Circle( int radius ) ;
    void area( const Circle& c ) ;
    ~Circle( int i ) ;
  private:
    int x, nr ;
}
```

```
// in main.cpp…
Circle c1;
```

Name: ____________________________________ Student Number________________

2. **Makefiles**
Consider the following makefile:

```
kernel: main.o Process.o Scheduler.o IO.o
        g++ main.o Process.o Scheduler.o IO.o -o kernel

Process.o: Process.h Process.cpp
        g++ -c Process.cpp

Scheduler.o: Scheduler.h Scheduler.cpp Process.h
        g++ -c Scheduler.cpp

IO.o: IO.h IO.cpp
        g++ -c IO.cpp
```

And consider the directory in which the makefile resides, which contains the following files with the indicated modification times:

```
19:44 IO.cpp
19:48 IO.h
19:46 IO.o
19:45 kernel
19:35 main.cpp
19:43 main.o
19:36 Makefile
19:43 Process.cpp
19:49 Process.h
19:47 Process.o
19:31 Scheduler.cpp
19:41 Scheduler.h
 9:45 Scheduler.o
```

What commands are executed and in which order when "make" is run?

Answer:

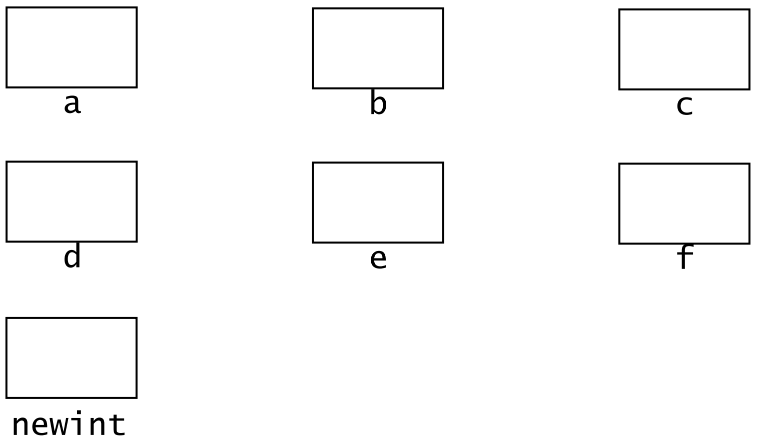
Name: ______________________________________ Student Number_________________

3. Memory and Pointers

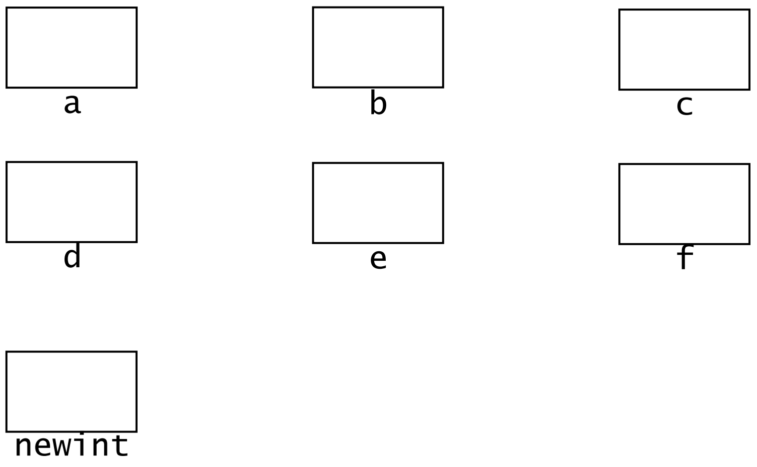
Assume that the following code will compile and run properly:

```
int a = 6;
int *b = &a;

int *
foo( int **c )
{
   (**c)++;
   *c = b;
   int *d = new int;
    *d = 10;
                         // Point #1
    return d;
}

int
main()
{
    int e = 7;
    int *f = &e;

    f = foo( &f );
    (*f)++;
                         // Point #2
    return 0;
}
```

a. Complete the following diagram by showing the values of variables and/or pointers <u>when execution reaches the point labeled "Point #1"</u>. For an integer variable, simply show the integer value inside the corresponding box. For a pointer, indicate the value of the pointer by drawing an arrow from the box corresponding to the pointer to the box corresponding to the variable the pointer points to.



b. Complete the following diagram by showing the values of variables or pointers when execution reaches the point labeled "Point #2". For an integer variable, simply show the integer value inside the corresponding box. For a pointer, indicate the value of the pointer by drawing an arrow from the box corresponding to the pointer to the box corresponding to the variable the pointer points to. Cross out any variables, pointers or memory allocations that no longer exist.



c. While the program will run correctly as written, it contains a non-fatal memory allocation problem. Write the single line of code that will fix the error. Specify the line number in the program after which the line of code should be inserted.

Answer:

4. Arrays and Linked Lists

a. Given the following class and array declaration, how would you print out the age of the 10th person in the array?

```
classpersonClass
  {
    public:
      void setAge( intnewAge ) ;
      void setGender( char newGender ) ;
      void setSalary( float newSalary ) ;
      int getAge() ;
      char getGender() ;
    private:
      int age ;
      char gender ;
      float salary ;
  } ;

personClass people[100] ;
```

A: `cout<< people[10] ;`
B: `cout<< people[9] ;`
C: `cout<< people[9].age ;`
D: `cout<< people[9].getAge() ;`
E: `cout<< people[10].getAge() ;`

Answer:

b. Given the following declarations, how would you know if head is pointing to an empty list?

```
class Node  {
    public:
    int value ;
      Node *next ;
      …
}

Node *head ;
// other code here…
```

A: `if( head->next == NULL )`
B: `if( head == null )`
C: `if( head == NULL )`
D: `if( head->next = null )`
E: A and D
F: B and C

Answer:

c. What is the output of the following code?

```
 int* p1 ;
 int* p2 ;
p1 = new int ;
p2 = new int;
*p1 =1 1 ;
*p2 = 0 ;
p2 = p1 ;
cout << *p1 <<" " << *p2 << endl ;
```

A. 11 0
B. 0 11
C. 11 11
D. 0 0
E. 23 0

Answer:

d. What is wrong with the following code fragment?

```
int* p1 ;
 int* p2 ;
p1 = new int ;
p2 = new int ;
*p1 = 11 ;
*p2 = 0 ;
p2 = p1 ;
cout << *p1 <<" " << *p2 << endl ;
delete p1 ;
delete p2 ;
```

A. nothing
B. p1 and p2 both have the same value, so the delete p2 will cause an error
C. you have a memory leak.
D. B and C

Answer:

5. Classes

What does the following program print when running:

```
classmT {
  private:
    static int created ;
  public:
    mT() ;
   ~mT() ;
}

int mT::created = 0 ;

mT::mT() {
    mT::created++;
    cout << mT::created
         << ”created” << endl ;
}

mT::~mT() {
    mT::created--;
    cout <<mT::created
         << ”remaining” << endl;
}

mT g[2] ;
mT h ;

void testFunction() {
    mT t ;
    mT *p NULL ;
    cout << “testFunction”
         << endl ;
    p = new mT ;
    cout << “leaving” << endl ;
}

int main() {
    mT m ;
    mT*p = new mT ;

    cout << “main” << endl ;
    testFunction() ;
    cout << “back” << endl ;
    delete p ;
    cout << “exiting” << endl ;
    return( 0 ) ;
}
```

Answer:

Name: ________________________________ Student Number______________

6. Scope

Show the output of the following program in the box provided below. Show your work for partial credit.

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

char confuse( char c1 ) ;
voidmixup( char &c1, char c2, char c3 ) ;

char c4 = 'F' ;

int main() {
    char c1 = 'H' ;
    char c2 = 'C' ;
    char c3 = 'E' ;

    mixup( c2, c1, c3 ) ;

    {
        char c3 = 'G' ;
        cout << c1 << c2 << c3 << c4 << endl ;
    }

    cout << confuse( c4 ) << c1 << c2 << c3 << c4 << endl;
    return 0 ;
}

char confuse( char c1 ) {
    const char c2 = 'O' ;
    char c3 = c1 ;
    if ( c1 >= 'N' ) return c3 ;
    else             return c2 ;
}

void mixup( char &c1, char c2, char c3 ) {
    cout << c1 << c2 << c3 << c4 << endl ;
    c1 = confuse( c3 ) ;
    c2 = 'S' ;
    c3 = 'I' ;
    c4 = confuse( c2 ) ;
}
```

Output line 1:	
Output line 2:	
Output line 3:	
Output line 4:	

7. Functions and Parameters

a. Identify all errors in the code below and give a very brief explanation of the error:

```
int f( inti, int& j ) {
  …
  …
  return( j ) ;
}

  …
int y ;
int x = f( 1, y ) ;
int z = f( 1, 2 ) ;
```

b. Consider the start of the following class declaration:

```
class Foo {
    public:
      void x( Foo f ) ;
      void y( const Foo f ) ;
      void z( Foo f ) const ;
```

Which of the three member functions can alter the PRIVATE member variables of the Foo object which the function is being invoked on:

A: only x
B: only y
C: only z
D: two of the functions can alter the private member variables
E: all of the functions can alter the private member variables

Answer:

c. When should you use a `const` reference parameter?

A: whenever the data type might be many bytes large
B: whenever the data type might be many bytes large, the function changes the parameter within its body, and you do NOT want these changes to modify the actual argument
C: whenever the data type might be many bytes large, the function changes the parameter within its body, and you DO want these changes to modify the actual argument
D: whenever the data type might be many bytes large, and the function does not change the parameter within its body.

Answer:

d. When should a pointer parameter `p` be a reference parameter?

A: When the when the function changes `p` and you want the change to modify the actual pointer argument.
B: When the function changes `p`, and you do NOT want the change to modify the actual pointer argument
C: When the function changes `*p` and you want the change to modify the object that is being pointed at.
D: When the function changes `*p` and you do NOT want the change to affect the object that is pointed at
E: When the pointer points to a large object.

Answer:

e. Suppose you have the following function and variable declaration:

```
void goop( int z[] ) ;
void x[10] ;
```

which is the correct way to call the goop function with `x` as the argument?

A: `goop( x ) ;`
B: `goop( x[] ) ;`
C: `goop( x[10] ) ;`
D: `goop( &x ) ;`
E: `goop( &x[] ) ;`

Answer:

8. Dynamic memory allocation

Given the following program, how many instances of `MyClass` exist in memory when the program execution reaches POINT X?

```
MyClass *func1( MyClass a ) {
   MyClass b ;
   MyClass* c = new MyClass() ;
   return c;
}

void func2( MyClass d ) {
   MyClass e ;
   MyClass* f = new MyClass() ;
   MyClass *g = func1( e ) ;
    // POINT X
   Return ;
}

int main() {
   MyClass h ;
   MyClass* i = new MyClass() ;
   func2( h ) ;
   return 0 ;
}
```

Answer: [ ] objects exist(s) in memory when the program reaches **`POINT X`**.

9. I/O

Write a program that reads integers from the command shell (input) until either the word “end” is inputted or the end-of-file is encountered. Assume that each integer, or the word “end”, appears on a line by itself, and that no erroneous integers are inputted. When an integer is read, it is simply printed to the output followed by a new line. The main function is provided to you below, and you must complete the code in this function as indicated. Rely on `cin` and do not use the `getline()` function.

```
int main () {
   int x ;

   // complete code here











    return (0);
}
```