

**University of Toronto**  
**Faculty of Applied Science and Engineering**

Midterm – October, 2009

ECE244 --- Programming Fundamentals

Examiners: Courtney Gibson, Wael Aboelsaadat, 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 8 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.

2 point deduction for not doing this...

**Last Name:** \_\_\_\_\_

**First Name:** \_\_\_\_\_

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

*Do not write below this line:*

---

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

### 1. Warmup questions

- a. What “submit” command would you run in order to submit your Driver.cpp and Makefile from Lab #3?

Answer:

**submitece244f 3 Driver.cpp Makefile**

3 points

-1 point for each “typo”  
with a min. mark of 0

- b. Consider a program which contains the following line of code:

```
ofstream logFile = "../../myProgram.log" ;
```

What is the *absolute path* to the directory in which myProgram.log will be created, if you run this program from “/home/jskule/ece244/lab4/”?

Answer:

**/home/jskule/myProgram.log**

3 points all or nothing

- c. You have written a program consisting of two source files: main.cpp and library.cpp. What command(s) would you run in order to compile these two files into a single program called testProgram?

Answer:

**g++ -o testProgram main.cpp library.cpp**

**(There are quite a few other correct answers.)**

4 points.  
No partial marks.

**2. C++ I/O**

- a. Using only “cin”, “cout” and their related commands, write code to receive input from the user and print either “real” or “integer”, depending on whether the user entered a real number (e.g., 14.7) or an integer (e.g., 14). You do not need to write a complete function, and you can assume that the input is either a real number or an integer. You are not to use the peek function.

Answer:

```
double d ;

cin >> d ;
int i = d ;
if( d == i )
    cout << “integer” ;
else
    cout << “real” ;
```

5 points

- b. Using only “cin” and its related commands, write code to accept either a real number or the string “midterm” from the user. If the user enters anything else (for example “finalExam”) you should print “error” and ignore the rest of the line, up to the next ‘\n’ character.

Answer:

```
double d ;

cin >> d;
if( cin.fail() ) {
    char buf[100] ;
    cin.clear() ;
    cin >> buf ;
    if( strcmp( buf, “midterm” ) ) {
        cout >> “error” ;
        cin.ignore( ‘\n’, 1024 ) ;
    }
}
```

5 points

Basic ideas to look for:

- First read into double
- Check for fail()
- Clear error
- Read into char buf
- Compare to “midterm”

One point each

**3. Pointers**

a. What is the output of the following code fragment:

```
int v1=2, v2=-1, *p1, *p2 ;
p1 = &v1 ;
p2 = &v2 ;
p2 = p1 ;
cout << *p2 << endl ;
```

2 points  
All or nothing

Answer:  
**2**

b. What is the output of the following code fragment:

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

2 points  
All or nothing

Answer:  
**11 11**

c. What is wrong with the following code fragment:

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

3 points  
All or nothing

- 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:  
**D**

d. Which of the following statements correctly returns the memory from the dynamic array pointer p1 back to the system?

- A: delete [] p1 ;
- B: delete p1 [] ;
- C: delete \*p1 ;
- D: delete p1 ;

Answer:  
**A**

3 points  
All or nothing

#### 4. Arrays and Linked Lists

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

```
class personClass
{
    public:
        void setAge( int newAge ) ;
        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() ;

2 points  
All or nothing

Answer:  
**D**

- 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

2 points  
All or nothing

Answer:  
**C**

- c. Which of the following code fragment correctly inserts a new node containing num into a linked list after the node pointed to by afterMe?

```
void insert( Node *afterMe, int num) {
    // code goes here...
}
```

A: afterMe->link = new Node ;  
 afterMe->link ->number = num ;  
 afterMe->link->link = afterMe->link ;

B: Node \* tmp = new Node ;  
 tmp->number = num ;  
 afterMe->link = tmp ;  
 tmp->link = afterMe->link ;

C: Node \*tmp = new Node ;  
 tmp->number = num ;  
 tmp->link = afterMe->link ;  
 afterMe->link = tmp ;

D: Node \*tmp = new Node ;  
 tmp->number = num ;  
 afterMe->link = tmp ;  
 tmp->link = NULL ;

3 points  
 All or nothing

Answer:

**C**

- d. If there is a Node pointer named toDelete that points to a valid node in the list, which of the following statements would remove the node that follows toDelete from the list and returns that memory back to the system? Assume a node exists in the linked list after the node pointed to by toDelete.

A: toDelete->link = toDelete->link->link ;  
 delete toDelete->link ;

B: Node \*tmp = toDelte->link ;  
 toDelete->link = tmp->link ;  
 delete tmp ;

C: Node \*tmp = toDelete->link ;  
 toDelete->link = toDelete->link->link ;  
 delete tmp ;

D: all of the above

E: none of the above

F: A and B

G: B and C

3 points  
 All or nothing

Answer:

**G**

10 points  
-1 point for each error  
0 points minimum

## 5. Classes

What does the following program print when running:

```
class mT {
    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:

**1 created**  
**2 created**  
**3 created**  
**4 created**  
**5 created**  
**main**  
**6 created**  
**testFunction**  
**7 created**  
**leaving**  
**6 remaining**  
**back**  
**5 remaining**  
**exiting**  
**4 remaining**  
**3 remaining**  
**2 remaining**  
**1 remaining**

## 6. Makefiles

You are given the following four files:

```

firstClass.h
#ifndef __FIRSTCLASS__
#define __FIRSTCLASS__

class firstClass
{
    private:
        int value;

    public:
        firstClass();
};
#endif
    
```

```

secondClass.h
#ifndef __SECONDCLASS__
#define __SECONDCLASS__
#include "firstClass.h"

class secondClass
{
    private:
        firstClass object;
};
#endif
    
```

2 points for the ifndef/defines  
2 points for the include

```

firstClass.cpp

#include "firstClass.h"

firstClass::firstClass()
{
    value = 0;
}
    
```

```

main.cpp

#include "secondClass.h"
#include "firstClass.h"

int main()
{
    firstClass first;
    secondClass second;

    return (0);
}
    
```



- a) The “.h” header files are both missing some “#...” preprocessor instructions (i.e., “#endif”, etc). Please add, in the space provided above, the missing line(s) to both files so that the code compiles correctly. You can assume that the “.cpp” files are both correct as written.
- b) In the space below, write the “g++” commands required to compile firstClass.cpp and main.cpp into a program named myProgram.

2 points

All or nothing

**g++ -c firstClass.cpp**

**g++ -c main.cpp**

**g++ main.o firstClass.o -o myProgram**

**or**

**g++ main.cpp firstClass.cpp -o myProgram**

- c) Write a complete Makefile for compiling “myProgram” from the files provided, including all file dependencies. The minimal set of file should be compiled.

**myProgram: main.o firstClass.o**

**g++ main.o firstClass.o -o myProgram**

**main.o: main.cpp firstClass.h secondClass.h**

**g++ -c main.cpp**

**firstClass.o: firstClass.cpp firstClass.h**

**g++ -c firstClass.cpp**

4 points max

-1 point for each error

0 points min

## 7. Constructors

Consider the following class definition:

```
class A
{
    private:
        B b;
        B *bp ;
        int value ;
        ...
    public:
        ...
}
```

Write a copy constructor that does a deep copy in the space provided below:

Answer:

```
A::A( const A& source )
{
    if( source.bp == NULL )
        bp = NULL ;
    else
        bp = new B ( *source.bp ) ;
    b = source.b ;
    value = source.value ;
}
```

10 points.

### 8. Value and Reference Parameters

a. Identify all errors in the code below:

```
int f( int i, int& j ) {  
    ...  
    ...  
    return( j ) ;  
}  
  
...  
int y ;  
int x = f( 1, y ) ;  
int z = f( 1, 2 ) ;
```

Cannot pass an integer value by reference.

4 points.

b. Consider the following code fragment:

```
int a = 2 ;  
  
class A  
{  
    private:  
        int a ;  
        int b ;  
    public:  
        A() { a = 5 ; b = 10 ; }  
        void f( int& ) ;  
};  
A* ap = new A ;  
  
A::f( int& c ) {  
    int a = b + c ;  
    a += 5 ;  
    c = a + b + c ;  
}  
  
...  
int c = 15 ;  
ap->f( c ) ;  
cout << c << endl ;
```

6 points all or nothing

Answer:

**55**

What is the output?