

Midterm Exam

Sept 26, 2012

NAME: _____

OPEN BOOK, OPEN NOTES, CALCULATORS ALLOWED.

1. Matrix Calculator Assignment (25 Points)

- (a) Suppose that in your Matrix class implementation, you did not define or implement a copy constructor, assignment operator or destructor (in other words you just used the default implementation for each of these. What would the `matrix-calculator` program do when processing the matrix operations found in the sample `input.txt` given in the assignment? Explain your answer.

- (b) Suppose you implemented the destructor as shown below, but again did not implement either the copy constructor or assignment operator?

```
Matrix::~Matrix() // destructor
{
    delete [] elements; // Free the elements array
}
```

- (c) A correct implementation of the copy constructor and the assignment operator are nearly identical. State two differences in the implementation of these two operations.

2. What is printed by the program subroutines on the next pages? **(25 Points)**.

Hint: There are eight outputs.

```

1 // Subroutine parameter passing example.
2 // ECE2036, Fall 2012
3
4 #include <iostream>
5 using namespace std;
6
7 int Sub2( int arg0, int& arg1)
8 { // Subroutine with two args; does some manipulation of the arguments
9   // and them multiplies the results and returns the product.
10  return ++arg0 * arg1++;
11 }
12
13
14 int Sub1( int* pInt0, int* pInt1)
15 {
16   return (*pInt0++) * (*pInt1++);
17 }
18
19
20 int main()
21 {
22   // Two arrays for manipulating
23   int A[10] = { 0, 10, 20, 30, 40, 50, 60, 70, 80, 90 };
24   int B[10];
25
26   int* pAzero = A;
27   int* pAone = &A[1];
28   int sub1Return = Sub1(pAzero, pAone);
29   cout << "Sub1 Return is " << sub1Return << endl;
30   cout << "*pAzero is " << *pAzero << endl;
31   cout << "*pAone is " << *pAone << endl;
32   int sub2Return = Sub2(A[1], A[2]);
33   cout << "Sub2 Return is " << sub2Return << endl;
34   cout << "A[1] is " << A[1] << endl;
35   cout << "A[2] is " << A[2] << endl;
36   int* pA = A;
37   int* pB = B;
38   *pB++ = *pA++;
39   *pB++ = pA[2];
40   cout << "B[0] is " << B[0] << endl;
41   cout << "B[1] is " << B[1] << endl;
42 }

```

Program subroutines.cc

3. Constructors and destructors (25 Points)

In the program `q1.cc` attached, identify where (what line number) each of the default constructors, int constructors, copy constructors, and destructors are called for each class A and B. Specify which line of code causes each of the above and a brief explanation of why the constructor was called. Be sure to note that the addition operator for classes A and B are both defined differently and implemented differently. As an example of how to fill in the table, one entry for the A int constructor is filled in.

A Default Constructor	Line Number	Explanation
A int Constructor	Line Number	Explanation
	38	Declaration of local variable "a" with int argument
A Copy Constructor	Line Number	Explanation
A Destructor	Line Number	Explanation
B Default Constructor	Line Number	Explanation
B int Constructor	Line Number	Explanation
B Copy Constructor	Line Number	Explanation
B Destructor	Line Number	Explanation

```

1 // Code for ECE3090 midterm, QUESTION 1 - Constructors and Destructors
2
3 class A {
4 public:
5     A();           // Default constructor
6     A(int);        // int Constructor
7     A(const A&);  // Copy constructor
8     ~A();          // Destructor
9     A operator+(const A& rhs) const; // Addition operator
10 public:
11     int x;         // Single data member
12 };
13
14 A A::operator+(const A& rhs) const
15 {
16     A r(x + rhs.x);
17     return r;
18 }
19
20 class B {
21 public:
22     B();           // Default Constructor
23     B(int);        // int Constructor
24     B(const B&);  // Copy constructor
25     ~B();          // Destructor
26     B operator+(B rhs) const; // Addition operator
27 public:
28     int x;         // Single data member
29 };
30
31 B B::operator+(B rhs) const
32 {
33     return B(x + rhs.x);
34 }
35
36 int main()
37 {
38     A a(1);
39     B b(2);
40
41     a = a + a;
42     b = b + b;
43 }

```

Program constructors-destructors.cc

4. MBed programming (25 points)

A pushbutton is connected from p8 to ground on the mbed module. What exactly does the following program do?

```
1  #include "mbed.h"
2  DigitalOut a(LED1);
3  DigitalOut b(LED2);
4  DigitalOut c(LED3);
5  DigitalOut d(LED4);
6  DigitalIn pb(p8);
7
8  int main()
9  {
10     int i=8;
11     pb.mode(PullUp);
12     wait(.001);
13     while(1) {
14         while(pb==1) {}
15         while(pb==0) {}
16         if (i==0) i=8;
17         d = i & 1;
18         c = (i >> 1) & 1;
19         b = (i >> 2) & 1;
20         a = (i >> 3) & 1;
21         i = i>>1;
22     }
23 }
```

Program mbed.cc