MECH 215 Programming for Mechanical and Industrial Engineering Final Exam
Section T/XX Instructors: Gordon / Obuchowicz
Date: Dec. 19, 2009, 9 am - 12 noon

- One double sided formula sheet and a calculator are permitted.
- There are 7 questions worth equal marks.
- Please **attempt any 5** questions.
- If more than 5 questions are attempted the first 5 questions will be marked and the **additional questions will not be marked**.

Question #1

Write a complete C++ program (including headers and the main function) which will generate the following output:

```
1 x 8 + 1 =  9
12 x 8 + 2 =  98
123 x 8 + 3 =  987
1234 x 8 + 4 =  9876
12345 x 8 + 5 =  98765
123456 x 8 + 6 =  987654
1234567 x 8 + 7 =  9876543
12345678 x 8 + 8 =  98765432
123456789 x 8 + 9 =  987654321
```

Your program MUST use a loop in order create the output. The loop itself must consist of C++ statements involving the mathematical operators '*' and '+'. NO points will be awarded for answers of the form:

```
cout << "1 x 8 + 1 =  9" << endl;
cout << "12 x 8 + 2 =  98" << endl;
etc.
```

Hint : 12 = 1 * 10 + 2
123 = 12 * 10 + 3
1234 = 123 * 10 + 4
12345 = 1234 * 10 + 5
123456 = 12345 * 10 + 6
etc.

Question #2

a) The pointer type determines how many bytes it is incremented by the ++ operator. True or false ?

b) What will be stored in the variables x and p after execution of the following lines of code ? Assume that variable x is stored in a memory location with the address 1000.

```
int x = 5;
int *p;

p = &x;
*p = 2;
p = p + 1;
```

c) What is an uninitialized (wild) pointer and what types of problems can it cause ? For example,

```
int *p;
*p = 1;
```

d) What is a NULL pointer and what is it used for ? Give an example.

e) What is the difference between call by reference and call by value function parameters ? Give an example.

f) What are the advantages of dynamic arrays versus static arrays ?

Question #3

Write a function string_convert(int n, char str[]) that converts a positive integer n into an array of characters (i.e. a string) called str. Note you cannot use string library functions in this program and you cannot use the math functions log10 or log. For example,

n = 2035

will result in:

str[] = { ‘2’,’0’,’3’,’5’,’\0’ };

note: the end of str should have the NULL character ‘\0’

Hint:

• if we count digits from right to left (least to most significant), the following expression can be used to calculate the i-th digit in n:

((int)(n/pow(10,i-1))) % 10

• for example, if n = 12345, the second digit (i=2) is 4

• the expression n/pow(10,i-1) will be less than 1.0 if i is larger than the number of digits.

• it might prove useful to define an array for the digit characters ‘0’, ‘1’, etc.

Question #4

a) Develop a structure (or class if you prefer) called **Text** that contains a dynamic 1D character array with 1000 elements. The member variables are given by:

int N; // length of the character array not including the termination character '\0'
char *p; // a pointer to the dynamic character array

Develop a set of member functions that includes:

b) A constructor that takes the character array (string variable) file_name as an argument. The constructor will allocate the dynamic array and then read the character array (string variable) in the file to initialize it. It will also initialize N.

c) A destructor.

d) A function called save(file_name) that saves the character array in **Text** to a file where the file name is stored in the character array file_name.

Each member function should print out a message and return if there is an error. Note you cannot use string library functions in this program.

Question #5

a) Develop a structure (or class if you prefer) called **Matrix** that contains the following member variables:

int N; // number of rows
int M; // number of columns
double *p; // a pointer to a dynamic 1D array of doubles

The Matrix structure/class is used to store a 2D array of doubles with indices starting at one (i.e. the rows go from 1 to N and the columns go from 1 to M). However, a dynamic 1D array of doubles is used to internally store the elements of the 2D array with each row stored in sequence. For example, the 2D array below would be stored as follows: // k = j + (i-1)*3

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix}$$

Note that the reason for using a 1D array for internal storage of a 2D array is because the new operator only creates a dynamic 1D array.

Develop a set of member functions that includes:

b) A constructor that takes as arguments n (number of rows) and m (number of columns). It then dynamically allocates a 1D array just large enough to store the 2D array. Finally, it initializes all the elements to zero and sets N and M to appropriate values.

c) A destructor.

d) A member function e(i,j) that returns the value of the i-th row and j-th column. It also checks to see if (i,j) is out of range and returns -1.0 if that occurs. For example, e(1,2) should return 2 for the given example above and e(2,3) should return 6.

Each member function should print out a message and return if there is an error.

Question #6

Give the output produced by the following program:

```
#include <iostream>
#include <string>

using namespace std;

void print(int a[][2][2][2])
{
for(int i = 0 ; i < 2 ; i++ )
{
 for(int j = 0 ; j < 2 ; j++)
 {
   cout <<  a[1][0][i][j] << " " ;
 }
 cout << endl;
}
}

void whole_lotta_array( int a[][2][2][2], int first, int second )
{
 int temp[2][2][2];

 for(int i = 0 ; i < 2 ; i++)
  for(int j = 0 ; j < 2 ; j++ )
   for(int k = 0 ; k < 2 ; k++)
     temp[i][j][k] = a[first][i][j][k];


 for(int i = 0 ; i < 2 ; i++)
  for(int j = 0 ; j < 2 ; j++ )
   for(int k = 0 ; k < 2 ; k++)
     a[first][i][j][k] = a[second][i][j][k];

 for(int i = 0 ; i < 2 ; i++)
  for(int j = 0 ; j < 2 ; j++ )
   for(int k = 0 ; k < 2 ; k++)
     a[second][i][j][k] = temp[i][j][k];

}
```

```
int main()
{

int array[3][2][2][2] = {  1,2,
                           3,4,
                           5,6,
                           7,8,
                           9,10,
                           11,12,
                           13,14,
                           15,16,
                           17,18,
                           19,20,
                           21,22,
                           23,24 };

print(array);

whole_lotta_array(array,1,0);

cout << "After whole_lotta_array it looks like: " << endl;

print(array);

return 0;

}
```

Question #7

Give the output produced by the following program:

```
#include <iostream>
#include <string>

using namespace std;

int main()
{

int keith;
int mick;

keith = 1;

// keith mick woody
// 1        X     X
// 1        X     2
// 1       11    2
//

for(int woody = 2 ; woody <= 10 ; woody++)
{
 mick = (keith * 9) + woody;
 cout <<  keith << " x 9 + " << woody << " = " << mick << endl;
 keith = (keith * 10)  + woody;
}

return 0;

}
```