

Name: *Exam Key*

Exam 2

CS151

This exam consists of 15 problems. You have until 12:15pm. The code used in this exam should be error free except for the problems in section B.

A. What do the following code snippets print? (5 points each)

1.

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

int foo( int x, int y );

int main()
{
    int x = 9;
    int y = 10;

    cout << x << " " << y << endl;
    cout << foo( x, y ) << endl;
    cout << x << " " << y << endl;

    return 0;
}

int foo( int x, int y )
{
    return x*=y;
}
```

Ans: *9 10*  
*90*  
*9 10*

2.

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

int foo( int &x, int &y );

int main()
{
    int x = 9;
    int y = 10;

    cout << x << " " << y << endl;
    cout << foo( x, y ) << endl;
    cout << x << " " << y << endl;

    return 0;
}

int foo( int &x, int &y )
{
    return x*=y;
}
```

Ans: *9 10*  
*90*  
*90 10*

3.

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

```
int main()
{
    int a = 9, b = 10;
    int *x, *y;

    y = &b;
    x = y;

    cout << a << " " << b << " " << *x << " " << *y << endl;
    return 0;
}
```

Ans:

9 10 10 10

4.

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

```
int main()
{
    int a = 9, b = 10, size = 4;
    int c[] = {1,2,3,4};

    c[3] = a;
    c[2] = b;

    for ( int i = 0; i < 4; i++ )
    {
        cout << c[i] << " ";
    }
    cout << endl;

    return 0;
}
```

Ans:

1 2 10 9

5.

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

```
int main()
{
    int col_size = 3, row_size = 2;
    int c[6][5];

    for ( int i = 0; i < 6; i++ )
    {
        for ( int j = 0; j < 5; j++ )
        {
            c[i][j] = i*j;
        }
        cout << endl;
    }

    for ( int m = 0; m < 5; m++ )
    {
        cout << c[m][m] << " ";
    }
    return 0;
}
```

Ans:

0 1 4 9 16

B. Find the logic or syntax errors in each of the following programs (there may be more than one). Circle the syntax errors you find. Describe the logic errors you find. (5 points each)

1. Find the syntax errors:

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

int main()
{
    int x = 9, y = 11;

    int *xPtr = NULL, yPtr = NULL;

    xPtr = x&;

    cout << (xPtr*) << " " << (*x) << endl;

    xPtr = &y;

    cout << (xPtr*) << endl;

    return 0;
}
```

2. Find the syntax errors:

```
void foo(&int x = 1, &int y = 1, &int z);

int main()
{
    int x = 0;
    for (int i = 0; i < 100; i++)
    {
        x = foo(i, i+1, i+2);
        cout << x << endl;
    }
    return 0;
}

void foo(&int x, &int y, &int z)
{
    return (x + y)(2.0*z);
}
```

*Handwritten notes:*  
 i; x and y have default values so must z  
 need an operator

\* Note:

reference parameter (int x) can't have default values unless they are declared const. You were not expected to know that for this exam.

3.

Find the syntax errors:

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

```
template<typename T> T foo( T x, T y )
```

```
int main()
```

```
{
    foo(3,0);
    return 0;
}
```

*needs two arguments*

```
template <typename T>
```

```
T foo( T x, T y )
```

```
{
    T z;
    return (x + y)/z;
}
```

*z uninitialized, possible logic error but not a syntax error*

4.

Find the logic error:

```
#include <iostream>
using namespace std;
int recur( int i );
```

```
int main()
```

```
{
    for ( int x = 0; x <= 100; x++ )
    {
        cout << recur( x + 1 ) << " ";
    }
    return 0;
}
```

```
int recur( int i )
```

```
{
    if ( i != 100 )
    {
        i = recur( i + 1 );
    }
    return i;
}
```

Ans:

*when x in main reaches 100 the function recur goes into an infinite loop since the condition for its base case is never met.*

5.

Find the logic error:

```
#include <iostream>
using namespace std;
int* bar( int x );

int main()
{
    int* array = NULL;
    for ( int i = 0; i < 100; i++ )
    {
        array = bar( i );

        for ( int i = 0; i < 100; i++ )
        {
            cout << array[i] << " ";
        }

        cout << endl;
    }
    return 0;
}

int* bar( int x )
{
    int a[100] = {0};
    a[x] = 1;
    return a;
}
```

Ans: *a is a pointer to a local set of variables (the array). The local array will return its memory when the function bar finishes. Bar will return a pointer to unallocated space.*

C. Write short but complete programs to do each of the following:  
(10 points each)

1. Write a *function* which takes a 500 integer array as its argument and returns the average of those integers. You only need to provide the *function definition*. Don't try to provide the values.

```
float average (int array[])  
{  
    float sum = 0.0;  
    for (int i=0; i < 500; i++)  
    {  
        sum += array[i];  
    }  
    return sum/500.0;  
}
```

2. Write a *function* that uses *iteration* to determine whether a string entered by the user is a palindrome. You only need to provide the *function definition*.

```
bool is_palindrome (char * input)
{
    int length = strlen(input);
    for (int i=0; i < length / 2; i++)
    {
        if (input[i] != input[length-i-1])
        {
            return false;
        }
    }
    return true;
}
```

3. Write a *function* to find the largest value in an array of 1000 integers. You only need to provide the function definition. Don't try to provide the values.

```
int largest (int input[])  
{  
    int max = input[0];  
    for (int i=1; i < 1000; i++)  
    {  
        if (max < input[i])  
        {  
            max = input[i];  
        }  
    }  
    return max;  
}
```

4. Write a *complete program* that uses iteration to put the reverse of a *string constant* (string literal) less than 101 characters long into a new character array.

```
int main ()
{
    char string_constant [] = "This is less than 101 chars";
    char new_array [101] = { '\0' };

    for (int i = 0; new string_constant [i] != '\0'; i++)
    {
        new_array [i] = string_constant [i];
    }

    return 0;
}
```

5. Write a complete program which uses a recursive function to concatenate two strings of length 5.

```
void recur_concat (char* array1, char* array2,  
                  int size, int counter = 0);
```

```
int main ()  
{
```

```
    char array1 [10] = { '1', '2', '3', '4', '5' };
```

```
    char array2 [5] = { '6', '7', '8', '9', '0' };
```

```
    recur_concat (array1, array2, 5);
```

```
    return 0;
```

```
}
```

```
void recur_concat (char* array1, char* array2,  
                  int size, int counter)
```

```
{
```

```
    if (counter < size) // recursive step
```

```
    {
```

```
        array1 [size + counter] = array2 [counter];
```

```
        recur_concat (array1, array2, size, ++i);
```

```
    }
```

```
    else // base case
```

```
    {
```

```
        array1 [size + counter] = '\0';
```

```
    }
```

```
}
```