COSC 120 **Exam 1 Key** Spring 2012

Part 1: Short Answer (5 Points Each)

1. What are the main differences between a compiler and an interpreter? – A compiler translates the high-level language program into machine language that can then be run directly on a computer. An interpreter, on the other hand, converts the program line by line into commands that are then run. The interpreter does not convert the entire program to machine language.

2. What are the three main types of programming errors and give a short explanation of how you find these errors.
   a) **Syntax Errors** – These are found by the compiler during the compilation process.
   b) **Run-Time Errors** – This is when you give the computer an instruction that cannot be completed, such as, division by 0. These are found during the testing stage.
   c) **Logic Errors** – These are when there are no syntax or run-time errors but the program is simply not doing what you wish it to do. These are also found during the testing phase.

3. What are the 4 types of decision structures?
   a) If Statement
   b) If Else Statement
   c) If Else if statement
   d) Switch Statement

4. What are the 3 types of loops?
   a) For Loop
   b) While Loop
   c) Do-While Loop

5. What does a #include statement do? – This statement loads in specific functions that are not loaded by default.

6. What is the difference between void functions and value returning functions? What is needed in the code for value returning functions? – A void function does not return a value through the function name. A value returning function does and in a value returning function you need to specify the value that is returned with a `return` statement.

7. What is the difference between pass by reference and pass by value? Discuss both the syntactical and memory usage differences. – If the parameter is passed by value then it is only the value that is passed to the function. With this type of passing, if a value parameter is changed in the function its value in the calling function is not changed. If the parameter is passed by reference then the memory location is passed to the function. In this case if the value is changed in the function it is also changed in the calling function. The pass by value scenario uses two different memory locations, one for the function and one for the calling function and in the pass by reference situation there is only one shared memory location. In pass by reference you need to place an & in front of the parameter name in the parameter list.

Part 2: Program Traces (10 points Each)

1. For each of the inputs in the box on the right give the output of the following program.

```
#include <iostream>

using namespace std;

int main()
{
    int n;
    cin >> n;

    while (n > 0)
    {
        if (n % 3 == 0)
            n = n/3;
        else if (n % 10 == 0)
            n += 2;
        else if (n % 2 == 0)
            n = n/2;
        else
            n--;

        cout << n << "  ";
    }

    return 0;
}
```

```
21
7  6  2  1  0


25
24  8  4  2  1  0


77
76  38  19  18  6  2  1  0


1
0
```

2. For each of the inputs in the box on the right give the output of the following program.

```
#include <iostream>

using namespace std;

void fct1(int &n, int &m, int &k)
{
    int temp = n;
    n = m;
    m = k;
    k = temp;
}

int fct2(int n, int m, int k)
{
    if ((n < k) && (n < m))
    {
        if (k < m)
            return k;
        else
            return m;
    }
    else if ((m < k) && (m < n))
    {
        if (k < n)
            return k;
        else
            return n;
    }
    else
    {
        if (m < n)
            return m;
        else
            return n;
    }

    return 0;
}

int main()
{
    int n1, n2, n3;
    cin >> n1 >> n2 >> n3;

    while (fct2(n1, n2, n3) < 10)
    {
        cout << n1 << "  " << n2;
        cout << "  " << n3 << endl;
        n1 *= 2;
        fct1(n1, n2, n3);
    }
    return 0;
}
```

```
5   3   1

5  3  1

3  1  10

1  10  6

10  6  2

6  2  20


6   3   4

6  3  4

3  4  12

4  12  6

12  6  8

6  8  24


1   3   3

1  3  3

3  3  2

3  2  6

2  6  6

6  6  4

6  4  12
```

3. For each of the inputs in the box on the right give the output of the following program.

```
#include <iostream>

using namespace std;

string fct1(string &s, int t, int r)
{
    string q = "";

    if (s.length() > t+r)
        q = s.substr(t, r);

    if (s.length() > t)
        s = s.substr(t);
    else
        s = "";

    return q;
}

int main()
{
    int n, m;
    string str1, str2 = "";

    cin >> str1;
    cin >> n >> m;

    while (str1.length() > 0)
    {
        str2 = str2 +
            fct1(str1, n, m);
    }

    cout << str2;

    return 0;
}
```

```
hdagfyakfgakfgaae 4 2
fyfgfg


qwertyuiopasdfghjklzxcvbnm 9 3
paslzx
```

Part 3: Coding (15 points Each)

1. Write a program that will take as input a positive integer n from the user, that is, the input could be 1 or 2 or 3 etc.. The program should calculate and output the sum of all the numbers from 1 to n, the sum of all the squares of the numbers from 1 to n and the sum of all the cubes of the numbers from 1 to n. That is, $1+2+3+...+n$, $1^2 + 2^2 + 3^2 + \ldots + n^2$, and $1^3 + 2^3 + 3^3 + \ldots + n^3$. The program should continually ask the user for another number until the user inputs -1. The calculation of each of the three sums should be done in three different value returning functions, each of those functions should take in the single parameter n.

```
#include <iostream>

using namespace std;

int Sum(int n)
{
    int sum = 0;
    for (int i = 1; i <= n; i++)
        sum += i;

    return sum;
}

int SumSq(int n)
{
    int sum = 0;
    for (int i = 1; i <= n; i++)
        sum += i*i;

    return sum;
}

int SumCube(int n)
{
    int sum = 0;
    for (int i = 1; i <= n; i++)
        sum += i*i*i;

    return sum;
}

int main()
{
    int n;

    do
    {
        cout << "Input n: ";
        cin >> n;
        if (n > 0)
            cout << "Sum = " << Sum(n) << "   Sum Squares = " << SumSq(n)
            << "   Sum Cubes = " << SumCube(n) << endl;
    }
    while (n > 1);

    return 0;
}
```

2. Write a program that will take an input of a positive integer number from the user, say n. Repeat the following process until n is equal to 1. If n is even replace n by n/2 and if n is odd replace n by 3n+1. The program must print out the sequence along with a count of the number of numbers in the sequence.

```
#include <iostream>

using namespace std;

int main()
{
    int n, count = 1;

    cout << "Input n: ";
    cin >> n;

    cout << n << "  ";
    while (n != 1)
    {
        count++;

        if (n % 2 == 0)
            n = n/2;
        else
            n = 3*n+1;

        cout << n << "  ";
    }

    cout << endl;
    cout << "Length: " << count;

    return 0;
}
```

3. Write a program that will take in an entire line as a string and count the number of occurrences of the character “t”, for example the input of “Take the tenth torch from the castle hall.” would return 7.The program should use a value returning function CountT that takes in the string as a parameter and returns the number of t's.

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

using namespace std;

int CountT(string str)
{
    int count = 0;

    for (int i = 0; i < str.length(); i++)
        if (str[i] == 't' || str[i] == 'T')
            count++;

    return count;
}

int main()
{
    string str;

    cout << "Input String: ";
    getline(cin, str);

    cout << "Count = " << CountT(str) << endl;

    return 0;
}
```