

Prelab 6 Reading

While Loops

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

int main()
{
    // Here is an example of a simple while Loop
    // while ( <conditional statement> )
    // {
    //     // statements
    // }

    char ans = 'y';
    cout << "Entering while Loop (enter 'n' to quit loop)" << endl;

    while (ans != 'n')
    {
        cout << "rock beats scissors" << endl;
        cin >> ans;
    }

    return 0;
}
```

Similar to a for loop, the while loop will only execute when the conditional statement is true, and it will continue to loop until the condition becomes false.

In the following example it is impossible to tell how many times it will execute. It depends on what the user inputs.

Do-While Loops

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

int main()
{
    // Here is an example of a simple do-while Loop
    //
    // do{
    //     // statements;
    // } while ( <conditional statement> );

    char ans = 'y';
    cout << "Entering do-while Loop (enter 'n' to quit loop)" << endl;

    do
    {
        cout << "Rock beats Scissors." << endl;
        cin >> ans;
    }
    while (ans != 'n');

    return 0;
}
```

While Vs. Do-While Loops

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

int main()
{
    cout << "While loop vs. do-while loop" << endl;

    //while
    while(false)
    {
        cout << "WHILE LOOP WIN" << endl;
    }

    //do-while
    do
    {
        cout << "DO-WHILE LOOP WIN" << endl;
    }
    while(false);

    return 0;
}
```

There is a subtle difference between a while loop and a do-while loop. A while loop tests its condition before entering the loop, whereas a do-while loop will test after the code inside the loop has executed. A do-while loop will always execute its code at least once, whereas a while loop may never run the code inside the loop.

Infinite Loops

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

int main()
{
    // Similiar to for loops, it is possible to create infintite loops with
    // while and do-while loops. Why is the following an infinite loop?

    int choice, i;
    cout << "\nEnter a negative number to create an infinite loop.\n";
    cin >> choice;
    i = 0;
    while (i > choice)
    {
        cout << "In infinite loop" << endl;
    }

    return 0;
}
```

Nested Loops

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

int main()
{
    // While and do-while loops can be nested. How many times will the 'cout'
    // execute? This is also how to implement a while loop to iterate a specific
    // number of times.

    const int SIZE2 = 3;
    int k;
    int j;
    int count_1 = 0;
    cout << "\nNested Loops" << endl;

    k = 0;
    while (k < SIZE2)
    {
        j = 0;
        while (j < SIZE2)
        {
            count_1++;
            cout << "k : " << k << " j : " << j
                << " count : " << count_1 << endl;
            j++;
        }
        k++;
    }

    cout << endl;

    // Would a FOR loop be better suited for this task?

    return 0;
}
```

Practical Example

```
// This program finds the factorial of any number below 20.
// The program will loop until it is told to terminate.

#include <iostream>
using namespace std;

int main() {
    int selection, g;
    long long int total; // can hold very large positive integers
    char menu;
    do {
        cout << "Welcome to my factorial calculator!\n";
        cout << "Please enter a positive number to find it's factorial!\n";
        cin >> selection;

        if (selection == 0) // if selection is equal to 0 return true (!0 = 1){
            cout << "0! = 1\n";

        }
        else if (selection > 20){
            cout << "The number entered is too large.\n";

        }
        else if(selection > 0){
            g = 2;
            total = 1;
            while (g <= selection){
                total *= g;
                g++;
            }
            cout << selection << "! = " << total << endl;

        }
        else {
            cout << selection << "! = undefined\n";
        }

        cout << "Would you like to calculate another number? ('y' or 'n')\n";
        cin >> menu;
    }
    while(menu != 'n');
    cout << "PROGRAM TERMINATED\n" << endl;

    return 0;
}
```