ENG 101: Day 4

Wednesday --12/1/05
Functions, review, and maybe arrays

Administrative

- GSI office hours updated again
  - Paul's Thursday hours have moved
    - Were 11:30-1:30. Now 6:30-8:30.
  - Did this so more folks could make it.

Administrative

- P0 pushed back until Wednesday of next week
  - GSIs (and I) concerned that would be too hard to have done by Wednesday.
  - This has caused fairly major changes in the class schedule.
    - See the webpage.
  - I think we will manage to stay on track now.

Due soon

- HW0 is due tomorrow by noon.
  - Box directions on website.

Where are we?

- We've done:
  - assignment, if/else, while loops, double vs. int, input, output.
- We've started on:
  - Functions
- Today we will talk about:
  - More functions, review some old code, and perhaps start on arrays.

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

int factorial (int value)
{
    int i=1;
    int fact=1;

    while(i<value)
    {
        i++;
        fact=fact*i;
    }
    return(fact);
}
```

```cpp
main()
{
    int max, a;

    cout << "Enter a non-negative integer less than 20 ";
    cin >> max;

    a=factorial(max);
    cout << max << " factorial is equal to " << a << endl;

    if(max<19)
    {
        max=max+1;
        a=factorial(max);
        cout << max << " factorial is equal to " << a << endl;
    }
}
```

And some problems with our code

- One icky thing is that we use the value 20 in two different places without explaination.
  - Well really 20 and 19
  - These are called "magic values" or "magic numbers" because the reader has no clue where they came from or if they are connected.
- The idea was that since the int type can only represent certain ranges, at some point the value of n! is too big.
  - When is that?

```
main()
{
    const int MAX_FACTORIAL=20;
    int max, a;


    cout << "Enter a non-negative integer less than " <<
            MAX_FACTORIAL << endl;
    cin >> max;

    a=factorial(max);
    cout << max << " factorial is equal to " << a << endl;

    if(max<MAX_FACTORIAL-1)
    {
        max=max+1;
        a=factorial(max);
        cout << max << " factorial is equal to " << a << endl;
    }
}
```

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

// Finds the 2 roots of a polynomial.  "which" should be
// only 0 or 1.  Different values of which give you the
// different roots.  Doesn't work if imaginary roots.
double qroot (double a, double b, double c, int which)
{
    double inside, top, bottom;

    inside=b*b - 4*a*c;
    if(which==0)
        top=-b + sqrt(inside);
    else
        top=-b - sqrt(inside);
    bottom=2*a;

    return(top/bottom);
}
```

```
main()
{
    double n2coef, n1coef, n0coef;
    double root1, root2;

    cout << "Enter the n squared coefficient ";
    cin >> n2coef;
    cout << "Enter the n coefficient ";
    cin >> n1coef;
    cout << "Enter the constant coefficient ";
    cin >> n0coef;

    root1=qroot(n2coef,n1coef,n0coef,0);
    root2=qroot(n2coef,n1coef,n0coef,1);

    cout << endl << "The roots are " << root1 << " and "
        << root2 << endl;
}
```

```
#include<iostream>
#include<cstdlib>
using namespace std;
main(int argc, char * argv[])
{
    const int trials=9000000;
    double x1, y1;
    double distance;
    int count=0; // number of hits
    int i=0;
    double value;

    while(i<trials)
    {
        x1=(1.0)*rand()/RAND_MAX;  // rand() generates an int [0.0, RAND_MAX]
        y1=(1.0)*rand()/RAND_MAX;
        distance=x1*x1+y1*y1; // Square of distance from home.
        if(distance<1)
            count++;
        i++;
    }
    value=(4.0)*count/trials;

    cout << "count= " << count << endl;
    cout << "value= " << value << endl;
}
```
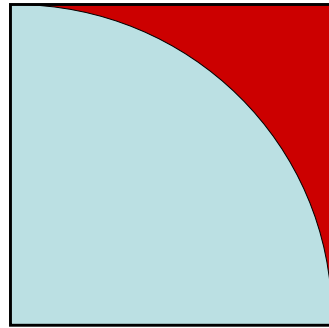
```cpp
#include<iostream>
using namespace std;


// Very silly program that prompts the user to enter a set of numbers.
// The user then enters one more number and the program tells the user
// which numbers from the original set are larger than his final number.
main()
{
    const int NUM=5;    // number of values user must enter
    int list[NUM];
    int i=0;
    int ans;            // number user enters.
    int any=0;          // set to 1 if any number is greater.

    cout << "You will be prompted to enter " << NUM << " numbers" << endl;
    while(i<NUM)
    {
        cout << "Enter a number ";
        cin >> list[i];
        i=i+1;
    }
    cout << "Now pick a number ";
    cin >> ans;
    i=0;
    cout << endl;
```

```cpp
    while(i<NUM)
    {
        if(ans<list[i])
        {
            cout << "Your number " << ans << " is less than " << list[i];
            cout << " from your list" << endl;
            any=1;
        }
        i=i+1;

    }
    if(any==0)
        cout << "Your number was greater than all numbers in the list" <<
endl;
    cout << endl << "Bye!" << endl << endl;

}
```