Basic Layout

Math and Computer Science

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

The Basics - Libraries

#include <iostream>	- basic input and output
#include <iomanip>	- formatting of input and output
#include <fstream>	- file input and output
#include <cctype>	- character type functions
#include <cstring>	- c string function
#include <cmath>	- math routines
#include "mylib.h"	- non system library for something

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The Basics - Comments

```
// Single line comment

/*
 * multiple line
 * comment
 */
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The Basics - Comments

Do not nest multiple line comments

```
/*
 * This is the
 * start
 /* of a multiple
  * line
  */                          ← this ends the first /*
 * comment
*/
```

some compilers handle nested, not worth taking chance.

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The Basics - Comments

Visual Studio does not handle nesting of comments

```
int main()
{
	/*
	cout << "Need to output" << endl;
	/* this is a comment */
	cout << "Many lines" << endl;
	*/
	cout << "Hello World" << endl;
	return 0;
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The Basics - Variable Declarations

- Start with _ or letter (A-Z,a-z)
- Can contain _, letters, or digits (0-9)
- Can not be a keyword (for, int, while, case, switch …)

- Are declared at the top of functions, not after the code starts
- This eliminates many of the scoping problems
- When you need a new variable, scroll to top and declare the variable
- Use the camelback naming Scheme (ex. int appleCount;)

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The Basics - Variable Declarations

- Global Variables are not allowed, I deduct a full letter grade for each global variable used.
- Global Constants are OK.

```
ifstream FIN;                    // not OK
const double PI = 3.14159;       // OK

int main()
{
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The Basics – Arithmetic statements

```
volume = ( 4 * PI * radius * radius * radius ) / 3.0;
```

- Use ( ) to override order of operations and make expression clearer.
- DO NOT over parenthesize the expression, learn your precedence.

```
volume = ((((  4 * PI  ) * radius) * radius) * radius  ) / 3.0);
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The Basics - Input and Output from Console

```
int x; double num; char ch; char multich[100];

cin >> x >> num >> ch >> multich;
cin.getline( multich, 80);

cout << x << " " << num << endl << ch
     << " " << multich << endl;
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The Basics – Namespaces

```
#include <iostream>

const double PI = 3.14159;
int main()
{
    double volume;
    double radius;

    // Prompt and get the radius
    std::cout << "Enter the radius"
              << " of the sphere: ";
    std::cin >> radius;

    // Calculate the volume
    volume = ( 4.0 * PI * pow( radius, 3 ) ) / 3.0;

    // display the resuts
    std::cout << "The volume of the sphere: "
              << volume << std::endl;

    return 0;
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The Basics – Namespaces

```
#include <iostream>

using std::cin;
using std::cout;
using std::endl;

const double PI = 3.14159;
int main()
{
    double volume;
    double radius;

    // Prompt and get the radius
    cout << "Enter the radius"
         << " of the sphere: ";
    cin >> radius;

    // Calculate the volume
    volume = ( 4.0 * PI * pow( radius, 3 ) ) / 3.0;

    // display the resuts
    cout << "The volume of the sphere: "
         << volume << endl;

    return 0;
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The Basics – Namespaces

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

const double PI = 3.14159;
int main()
{
    double volume;
    double radius;
    // Prompt and get the radius
    cout <<"Enter the radius of the sphere:";
    cin >> radius;

    // Calculate the volume
    volume = ( 4.0 * PI * pow( radius,3 )) / 3.0;

    // display the resuts
    cout << "The volume of the sphere: "
         << volume << endl;

    return 0;
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY