This is extra-credit homework to get more practice with programming, small steps at a time. Odd-numbered problems have answers on the next page, while you get credit for any even-numbered answers turned in. Practice doing both even and odd problems, using the answers as a guide afterward.

First, create a basic C++ program:

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

int main()
{
    return 0;
}
```

For the turn-in problems (even numbered), you can have one right after another in the same program. Use a comment to specify which part of the code belongs to which problem. Example:

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

int main()
{
    // Question 1.
    int number;

    // Question 2.
    char letter;

    return 0;
}
```

Submit the .cpp (C++ Source) file once you are done.

Questions: Functions that return a value (part 1)

Functions in programming can be similar to functions in algebra. Remember f(x)?

Points possible: 4

1.	Write a function that is equivalent to: $f(x)=3x+2$ Where the return type and the parameter are both integers.
2.	Write a function that is equivalent to: $f(x)=5x+9$ Where the return type and the parameter are both integers.
3.	Write a function that is equivalent to: $c(a,b)=\sqrt{(a^2+b^2)}$ Where the return type and parameters are floats.
4.	Write a function that is equivalent to: $f(x)=\sqrt{x^2}$ Where the return type and parameters are floats.
5.	Write a function called **GetName** that takes no parameters and has a return type of **string**. In the function, it returns a value of “Ada”.
6.	Write a function called **GetLocation** that takes no parameters and has a return type of **string**. In the function, it returns a value of “Kansas City”.
7.	Write a function called **FormatName** that takes two parameters: 1. firstName (string), 2. lastName (string). The return type of the function is also a **string**. In the function, it creates a string that is the combination of the first name and last name to look like “Hopper, Grace” and returns the formatted string.
8.	Write a function called **FormatLocation** that takes parameters: 1. city (string), 2. state (string), 3. country (string). The return type of the function is also a **string**. In the function, it creates a string that is the combination of the city, state, and country to look like “Kansas City, Missouri, USA”, and returns the formatted string.

Answers: Functions that return a value (part 1)

1.

```
int f( int x )
{
    return 3 * x + 2;
}
```

3.

```
float c( float a, float b )
{
    return sqrt( a*a + b*b );
}
```

5.

```
string GetName()
{
    return "Ada";
}
```

7.

```
string FormatName( string firstName, string lastName )
{
    string formatted = lastName + ", " + firstName;
    return formatted;
}
```

Questions: Functions that return a value (part 2)

Points possible: 3

9.	Write a function called **GetGrade**. Its parameter is a name (**string**), and the return type is a **char**. Within the function, there is an if statement. If name is “Ashley”, return the grade 'A'. If the name is “Beth”, return the grade 'B'. If the name is “Cathy”, return the grade 'C'. Otherwise, return an 'F'.
10.	Write a function called **GetAge**. Its parameter is the year (**int**), and the return type is an **int**. Within the function, subtract the current year from the **year** parameter and return this value. (We aren't worrying about month, just returning a general age.)
11.	Write a function called **GetChoice**. It has no parameters, and its return type is a **string**. Within the function, it displays the message, “What is your choice? ”. It has a variable called **choice**, which is a string. The user enters a value into **choice**, then **choice** is returned from the function.
12.	Write a function called **GetNumberChoice**. It has no parameters, and its return type is an **integer**.. Within the function, it displays the message, “What is your choice? ”. It has a variable called **choice**, which is an **integer**. The user enters a value into **choice**, then **choice** is returned from the function.
13.	Write a function called **GetYesOrNo**. It has no parameters, and its return type is a **string**. Within the function, it displays a message, “What is your choice? Yes or No: ”. It has a variable called **choice**, which is a string. The user enters a value into **choice**. While the choice is an invalid value (not yes or no), it will display an error message and have the user re-enter their choice. Once the user has entered a valid choice, return **choice** from the function.
14.	Write a function called **GetMenuChoice**. It has no parameters, and its return type is an **integer**. Within the function, it displays a message, “What is your choice? 1 to 4: ”. It has a variable called **choice**, which is a string. The user enters a value into **choice**. While the choice is an invalid value (less than 1, or greater than 4), it will display an error message and have the user re-enter their choice. Once the user has entered a valid choice, return **choice** from the function.

Answers: Functions that return a value (part 2)

9.

```
char GetGrade( string name )
{
    if      ( name == "Ashley" )    { return 'A'; }
    else if ( name == "Beth" )      { return 'B'; }
    else if ( name == "Cathy" )     { return 'C'; }
    else                            { return 'F'; }
}
```

11.

```
string GetChoice()
{
    string choice;
    cout << "What is your choice? ";
    cin >> choice;
    return choice;
}
```

13.

```
string GetYesOrNo()
{
    string choice;
    cout << "What is your choice? (Yes/No): ";
    cin >> choice;

    while ( choice != "yes" && choice != "no" )
    {
        cout << "Invalid choice. Try again: ";
        cin >> choice;
    }

    return choice;
}
```

Questions: Void Functions

Functions with a return type of “void” do not **return** any values. They can be useful for self-contained functionality, that do not need to return data back to the caller.

Points possible: 3

15.	Write a function called **DisplayMenu**. It has a return type of **void** and contains no parameters. All this function will do is display 4 options to the user: 1. Add Student, 2. Remove Student, 3. Update Class Name, or 4. Quit. The function only displays a menu, and is not responsible for any additional functionality.
16.	Write a function called **DisplayDirections**. It has a return type of **void** and contains no parameters. All this function will do is display 4 options to the user: N: North, S: South, E: East, W: West. The function only displays a menu, and is not responsible for any additional functionality.

17.	Write a function called **DisplayStats**. It has a return type of **void** and contains the following parameters: id (int), name (string), count (int), and averageGrade (float).
	Within the function, it will display the information with labels. For example:
	`SPRING 2016, CS 200`
	`ID: 1234  Total Students: 16  Average Grade: 3.9`

18.	Write a function called **DisplaySupportedLanguages**. It has a return type of **void** and contains the following parameters: language1 (string), language2 (string), and language3 (string). Within the function, it displays “SUPPORTED LANGUAGES:”, then displays the values of each of the parameters.
19.	Write a function called **CalculateSlope**. It has a return type of **void** and has no parameters. Within the function, create the following local variables: x1, y1, x2, y2, all floats. Ask the user to input x1, y1, x2, and y2, and store the input into these variables. Calculate `float m = (y1-y2)/(x1-x2)` , and display the result.
20.	Write a function called **CalculateSquare**. It has a return type of **void** and has no parameters. Within the function, create the following local variables: x, xSquared, both floats. Ask the user to enter the x. Calculate xSquared as `x*x`. Then display the result.

Answers: Void Functions

15.

```
void DisplayMenu()
{
    cout << "1. Add Student" << endl;
    cout << "2. Remove Student" << endl;
    cout << "3. Update Class Name" << endl;
    cout << "4. Quit" << endl;
}
```

17.

```
void DisplayStats( int id, string name, int count, float averageGrade )
{
    cout << "SPRING 2016, " << name << ", ID: " << id << endl;
    cout << "Total Students: " << count << endl;
    cout << "Average Grade: " << averageGrade << endl;
}
```

19.

```
void CalculateSlope()
{
    float x1, y1, x2, y2;

    cout << "Enter 1st Coordinate Pair x, y: ";
    cin >> x1 >> y1;
    cout << "Enter 2nd Coordinate Pair x, y: ";
    cin >> x2 >> y2;

    float m = (y1-y2)/(x1-x2);
    cout << "Slope is: " << m << endl;
}
```

Questions: Calling Functions (part 1)

For these, you need the functions that were written in the previous sections.

Points possible: 5

21.	Within `main()`, call the function from part 1. You will store the result in an integer named **y**, and you will pass in the value of **5** for x. Output the value of **y**.
22.	Within `main()`, call the function from part 2. You will store the result in an integer named **z**, and you will pass in the value of **5** for x. Output the value of **z**.
23.	Within `main()`, call the function from part 3. You will store the result in an integer named **cvalue**, and you will pass in the value of **2.5** for a and **3.0** for b. Output the value of **cvalue**. (You will need to `#include <cmath>`)
24.	Within `main()`, call the function from part 4. You will store the result in an integer named **xroot**, and you will pass in the value of **16** for x. Output the value of **xroot**. (You will need to `#include <cmath>`)
25.	Within `main()`, call the function **GetName** from part 5 in a **cout**, which should print the name to the screen.
26.	Within `main()`, call the function **GetFunction** from part 6 in a **cout**, which should print the name to the screen.
27.	Within `main()`, call the function **FormatName** from part 7 in a **cout**. Pass in “Stephanie” as the first name and “Shirley” as the last name.
28.	Within `main()`, call the function **FormatLocation** from part 8 in a **cout**. Pass in “Raytown” as the city, “Missouri” as the state, and “USA” as the country
29.	Within `main()`, create a variable called **grade**, of type **char**. Call the function **GetGrade**, pass in the value of “Cathy”, and store the result in the **grade** variable. Output **grade** to the screen.
30.	Within `main()`, create a variable called **age**, of type **int**. Call the function **GetAge**, pass in the value of 1985, and store the result in the **age** variable. Output **age** to the screen.

Answers: Calling Functions (Part 1)

21.

```
int main()
{
    int y = f(5);
    cout << y << endl;
    return 0;
}
```

23.

```
int main()
{
    float cvalue = c( 2.5, 3.0 );
    cout << cvalue << endl;
    return 0;
}
```

25.

```
int main()
{
    cout << GetName() << endl;
    return 0;
}
```

27.

```
int main()
{
    cout << FormatName( "Stephanie", "Shirley" ) << endl;
    return 0;
}
```

29.

```
int main()
{
    char grade = GetGrade( "Cathy" );
    cout << grade << endl;
    return 0;
}
```

Questions: Calling Functions (part 2)

For these, you need the functions that were written in the previous sections.

Points possible: 4

31.	Within `main()`, create a variable called **choice**, of type **string**. Assign **choice** equal to the **GetYesOrNo** function call from part 13. Afterwards, display the result of **choice** to the screen.
32.	Within `main()`, create a variable called **choice**, of type **int**. Assign **choice** equal to the **GetMenuChoice** function call from part 14. Afterwards, display the result of **choice** to the screen.
33.	Within `main()`, call the function **DisplayMenu** from part 15.
34.	Within `main()`, call the function **DisplayDirections** from part 16.
35.	Within `main()`, call the function **DisplayStats** from part 17. Pass in the values of 123, “CS 99”, 30, and 3.5.
36.	Within `main()`, call the function **DisplaySupportedLanguages** from part 18. Pass in the values of “English”, “Ido”, and “Toki Pona”.
37.	Within `main()`, call the function **CalculateSlope** from part 19. Run the program and enter the values 2.5, 1.0, 3.0, and 4.2. When you run the program, the resulting slope should be 6.4.
38.	Within `main()`, call the function **CalculateSquare** from part 20. Run the program and enter the value of 5. The result displayed should be 25.

Answers: Calling Functions (Part 2)

31.

```
int main()
{
    string choice = GetYesOrNo();
    cout << choice << endl;
    return 0;
}
```

33.

```
int main()
{
    DisplayMenu();
    return 0;
}
```

35.

```
int main()
{
    DisplayStats( 123, "CS 99", 30, 3.5 );
    return 0;
}
```

37.

```
int main()
{
    CalculateSlope();
    return 0;
}
```