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: Variables

Points possible: 4

1.	Declare a variable whose name is **amountOfCats**, and whose data type is an integer. Assign it a value of **20**
2.	Declare a variable whose name is **pricePerCat**, and whose data type is a float. Assign it a value of **49.99**
3.	Declare a variable whose name is **nameOfCat**, and whose data type is a string. Assign it a value of **PumpkinSparkles**
4.	Declare a variable whose name is **catLocation** and whose data type is a string. Assign it a value of **Kansas City**
5.	Declare a variable whose name is **grade** and whose data type is a character. Assign it a value of **A**
6.	Declare a variable whose name is **rating** and whose data type is a character. Assign it a value of **+**
7.	Declare a variable whose name is **success** and whose data type is a boolean. Assign it a value of **true**
8.	Declare a variable whose name is **done** and whose data type is a boolean. Assign it a value of **false**

Answers: Variables

1.	`int amountOfCats = 20;` int is the data type (integer), amountOfCats is the name of the variable, and we are assigning it a value of 20 with the assignment operator =.
3.	`string nameOfCat = "PumpkinSparkles";` the value of a string variable must be contained within double-quotes.
5.	`char grade = 'A';` the value of a char variable must be contained within single-quotes. Chars can only hold a single character – numbers, letters, symbols, etc.
7.	`bool success = true;` booleans can be set to true or false.

Questions: Output

Points possible: 5

9.	Output the text **Hello World** to the screen.
10.	Output the text **How are you?** To the screen.
11.	Output the text **I have** , then output the value of the **amountOfCats** variable, then output the text **cats**. Example output: `I have 20 cats.`
12.	Output the text **The price of a cat is** , then output the value of **pricePerCat**. Example output: `The price of a cat is 49.99`
13.	Output a single new line with no text.
14.	Output two new lines with one cout statement.

15.	Output the text **Cat Name:** , then output the value of **nameOfCat** . Then output a new line. Then, output the text **Cat Location:** , then output the value of **catLocation** .
	Example output:
	`Cat Name: PumpkinSparkles`
	`Cat Location: Kansas City`

16.	Output the text **Grade:** , then output the value of **grade** . Then output a new line. Then, output the text **Success:** , then output the value of **success** .
	(Note: success will show up as either a 1 or a 0, not true or false)
	Example output:
	`Grade: A`
	`Success: 1`

17.	With one cout statement, output the text **Rating:** With a second cout statement, output the value of **rating** on the same line.
18.	With one cout statement, output the text **Grade:** With a second cout statement, output the value of **grade** on the same line.

Answers: Output

9.	`cout << "Hello World";` the **cout** command stands for console-out. Use the stream operator <<. Strings must be contained within two double-quotes.

11.	`int amountOfCats = 20; // Problem 1`
	`cout << "I have " << amountOfCats << " cats.";`
	
	You can combine string literals (within " ") and variables. Any time you switch, you must put the stream operator << between them.

13.	`cout << endl;` The **endl** command stands for end-line.

15.	`cout << "Cat Name: " << nameOfCat << endl;`
	`cout << "Cat Location: " << catLocation << endl;`
	
	It looks nice when the program runs to end each **cout** line with an **endl** if you don't need the next **cout** statement to appear on the same line.

17.	`cout << "Rating: ";`
	`cout << rating;`
	
	If there is no **endl** at the end of a **cout**, any future **cout** statements will appear at the end of the first one.

Questions: Input

Points possible: 3

19.	Declare a variable named **age** whose data type is an integer. Output the text **Enter your age:** Then get the input from the user and store it in the **age** variable.
20.	Declare a variable named **balance** whose data type is a float. Output the text **How much money do you have:** Then get the input from the user and store it in the **balance** variable.
21.	Declare two variables of type float. Name them **number1** and **number2**. Output the text **Enter two numbers:** Get the input for two numbers and store them in **number1** and **number2**. Then, output the text **Sum:** And output the result of **number1 + number2**.
22.	Declare two variables of type float. Name them **x** and **y**. Output the text **Enter two numbers:** Get the input for two numbers and store them in **x** and **y**. Then, output the text **Product:** And output the result of **number1 * number2**.
23.	Declare a character variable and name it **userChoice** Output the text **Do you want to quit? Y or N:** Get the input and store it in the variable **userChoice** Output the text **You selected:** Output the value of **userChoice**.
24.	Declare a character variable and name it **input** Output the text **Which direction? N, E, S, or W:** Get the input and store it in the variable **input** Output the text **You selected:** Output the value of **input**.

Answers: Input

19.	`int age;`
	`cout << "Enter your age: ";`
	`cin >> age;`
	
	The **cin** command stands for console-in. The stream operator >> points away from the **cin** and towards the variable you are storing data in.

21.	`int number1, number2;`
	`cout << "Enter two numbers: ";`
	`cin >> number1 >> number2;`
	`cout << "Sum: " << number1 + number2;`
	
	You can declare multiple variables of the same type on one line, with the variable names separated by a comma. You can also get the input for two variables at once, by linking them together with the stream operator >>.

23.	`char userChoice;`
	`cout << "Do you want to quit? Y or N: ";`
	`cin >> userChoice;`
	`cout << "You selected: " << userChoice;`
	
	Use a generic name for variables. The user is inputting their **choice**. Later on, we will worry about whether they chose Y or N, by using if statements. Don't name the variable **yes** or **no**.