Control Structures, Cout and Cin
https://www.cplusplus.com/files/tutorial.pdf Pages (34-40, 23-24)

The if statement, checks the statement inside the () and then decides whether to do the stuff inside the {}. An if(){} can be followed by an else if(){}, which is an if that only checks if the first if(){} was false. An if(){} can also be followed by an else{}, which is the “backup” option if none of the other options were taken.

```
if(){
}
else if(){
}
else{
}
```

The while statement will repeat itself while the statement inside the () is true.

```
while(){
}
```

Statements inside the parens can use the following tests:

==	Equal to	!=	Not equal to
<	Less than	>	Greater than
<=	Less than or equal	>=	Greater than or equal

An example that prints every odd number from 0 to 100:

```
int count = 0;
while(count <= 100){
      if((count % 2) == 1){
            cout << count << endl;
      }
}
```

We have already used cout to print to the terminal, and with cin we will be able to take input from the user’s keyboard. Both cin and cout are special variables called streams that interact with parts of the computer, and other streams can be used to read files or make connections to the internet. cin and cout are covered in detail on pages 29-33 of the textbook.

For our simple purposes, we will just use the >> and << operators. Cout uses >> to insert our variables into the terminal, and cin uses << to pull input from the terminal and put it into variables.
To output:

```
cout >> variable;
```

To get inputs:

```
cin << variable;
```