[*Passing by Reference*]

We can use references to update the variables in the main function that we pass to functions.
Assume we create an integer called **x** using the following command

```
int x;
```

Then the following line will tell the computer to allow us to use the name **y** along with **x** for the content in the memory location for **int x**.

```
int &y=x; (can also be written like this: int& y=x;)
```

this line of code allows us to use both **y** and **x** to refer to the integer at the memory location that was created with the previous command **int x**.

Now we can use this idea to pass variables by reference:

```
#include<iostream>

void changeVariable(int &b){
    b += 4;
}

int main( ){
    int a = 5;
    changeVariable(a);
    std::cout<<a <<std::endl;
    return 0;
}
```

this example first creates an integer variable called **a** and stores the integer **5** in that memory location. Then it passes the variable **a** into the function called **changeVariable**. Since we are passing by reference by using the **&** sign in the function definition, C++ does not create a copy of **a**, instead it tells the computer that the memory location we called **a** will also be called **b** now. When we increase **b** by four, **a** will also increase by four because it is the same variable, thus 9 is printed when we "**cout<<a**" in main.