Variables and cin
author Andrew Lycas

so when u do math u know what a variable is, but on ur paper there is one type of variable, and this variable could hold anything if u wanted it to, usually u have it hold an unknown number that u eventually solve. In programming, variables are a little bit different

There are different types of variables : int, float, double, char, string are just to name a few.

Int is an integer, so it is a non decimal number ex. 5
float and double are both decimal numbers ex .654 or 8.34
char is a single character ex g or 4
string a group of chars ex hello or woo

the actual number ones have some huge range that if ur number is to big it wont get any bigger, but its irrelevant Just know that it is there.

U might ask, well y not just always just use floats or doubles? Well in a program every thing u create (thats essentially what u are doing when u assign a variable to the variable) takes up space. And in small programs there is no problem with making everything a double, but when u have a limited amount of space, like on the robot, u have save space. Also, it is good programming practice to make what u know will stay an int an int and the rest w/e.

Now in ur program, u have to declare the variable, and give it a value before u can use it. If u do not declare it, u cant use that variable later, and if u don't give it a value, the computer just automatically gives it a 0 as its value.

```
--
int myInteger ;
double thisDouble = 89.3;
--
```

above is an example of declaring the variables By putting int myInteger, u are telling the computer that when u refer to myInteger later, that myInteger is a variable, it is and integer, and it has a starting value of 0. The next line is doing the same thing, but it is a double instead of an int, and it is setting the starting value to 89.3 instead of 0.

put this example program on the next page into the complier and run it. You can see that the variables are defined in the beginning. And when it is run it is displaying the values of the variables Notice that when i used std::cout the variable name did not have “” around it. If there were quotes around it, it would display what was literally in the quotes, without them, the computer looks for the variable and displays its value.

--
```
#include "iostream.h"
#include "stdio.h"

int main()
{
	int myInteger=9;
	double myDouble = 9.9;


	std::cout<< myInteger << "," << myDouble  ;
	getchar();

	return 0;
}
```
--

also regarding the std::cout command, whenever u switch between “” and variables u need to put << between them. Just a rule that needs to be followed.

Well displaying variables that were hard coded is kinda useless in this application, so what about some user interaction..

the std::cin command is the opposite of cout. cin will take in stuff and u can store it as a variable

Put this code in for an example:
--

```
#include "iostream.h"
#include "stdio.h"

int main()
{
	int myInteger;
	double myDouble;


	std::cout<< "Give me an integer :";
	std::cin >> myInteger ;
	std::cout<< "\nGive me a double :";
	std::cin >> myDouble ;
	std::cout << "\n\n\n You entered the numbers " << myInteger << " and " << myDouble ;

	getchar();
	getchar();

	return 0;
}
```
--

Alright, this code is similar to the first set of code in that there are 2 variables and that they are eventually displayed. However a couple things in here we haven't mentioned here such as the \n and the cin's syntax.

“\n” is just a return. It makes space. Use it when u need to go to another line. For those that know html, its like the
 statement.

With the cin statement u need to have its >> going towards the variable They way i remember it and the way they probably intended it was that with cout the << are opening up and showing whats in the larger end of the <. With cin, the >> are going into the variable that is getting the information. From cin into the variable

Also with variables u can add them subtract them and other operations to them see the example:

int a = 6;
int b = 2;

a+b=8
a-b=4
a*b=12
a/b=3
a%b=0 (% is modulus, meaning if the two were divided, what the remainder would be)
a^b = 36 (a to the b power)

that is the basic jist of numerical variables Chars are used mainly for comparing things, and strings is a very complicated category.

One major difference between c and c++ is that in c u have to declare all variables before anything else in your function. In c++ u can declare them anywhere. This might seem not important because right now we are programming in c++, however just remember that the robot is programmed in c. which means if u declare a variable randomly in the robot code, u will get an error. But right now its ok, but keep that in the back of your mind when we get to programming the robot.

Next lesson: project 1: the calculator