Warlocks 1507

# FRC Programming in C++

Class 4

**Lockport Robotics**
**October 2023**

FIRST

# Important Concepts

| | | |
|---|---|---|
| Variables | Storage | x = 10; |
| Conditionals | Decisions | if( x == 10 ) |
| Loops | Repeated instructions | for(i=0; i<10; i++) |
| Functions | Small Tasks | x = GetEncoderValue(); |
| Classes | Objects | Motor leftMotor;<br>Motor rightMotor; |
| Class Members | variables and functions in a class | leftMotor->Set(0.7); |

# Two types of programmers

```
if( <condition> ){
    //Statements
}
```

These programmers:

- Live lonely pathetic lives
- Burn orphanages for fun
- Probably still live with parents at 30

```
if( <condition> )
{
    //Statements
}
```

These programmers:

- Are well respected in the community
- Rarely write bugs
- Make lots of money

# Two types of programmers – Part 2

Ones who use ***SPACES***

everything lines up
nice and neat
and is easy to read

Ones who use ***TABS***

nothing lines
up
correctly
and
is
very difficult to
read

# Basic Variable Data Types

| | |
|---|---|
| int | 4 byte integer number |
| float | 4 byte rational number |
| double | 8 byte rational number |
| char | 1 byte integer number |
| bool | 1 bit ( True/False ) |

# Math Operators

| | |
|---|---|
| + | Add |
| - | Subtract |
| * | Multiply |
| / | Divide |
| % | Modulus ( division remainder) |
| ++ | increment (+1) |
| -- | decrement (-1) |

Warlocks 1507

# Conditionals

IF-Then

```
if(  <test>  )
{
    //Runs if true
}
```

IF-Then-else

```
if(  <test>  )
{
    //Runs if true
}
else
{
    //Runs if False
}
```

Nested IF

```
if(  <test1>  )
{
    //Runs if true
}
else if( <test2> )
{
    //Runs if test1 false
    //but test2 true
}
else
{
    //Runs if all tests False
}
```

# Conditionals

## Logical Operators

- Results are always Boolean (True/False)

| | |
|---|---|
| == | Equal |
| != | Not Equal |
| > | Greater Than |
| >= | Greater  Than or Equal |
| < | Less Than |
| <= | Less Than or Equal |
| && | Logical AND |
| \|\| | Logical OR |
| ! | Logical NOT |

# Assignment vs Comparison

| Symbol | Meaning | Example | Results |
|---|---|---|---|
| = | Assignment | x = 2; | var x contains value 2 |
| == | Comparison | x == 2 | True if x is 2 else False |

# Loops

## While Loop

```
while( <test> )
{
    //Run while true
}
```

```
while( ReadGyro() < 45 )
{
    Turn();
}
```

```
while( count < 5 )
{
    if( onTarget == True )
    {
        count++;
    }
    else
    {
        count = 0;
    }
}
```

# Loops

## For Loop

```
for( <init>; <test>; <inc> )
{
    //Run while true
}
```

```
for( int i = 0; i < 10; i++)
{
    cout << "i= " << i << endl;
}
```

```
// Same as:
int i = 0;
while( i < 10 )
{
    cout << "i= " << i << endl;

    i++;
}
```

# Functions

- Allow us to structure our program in a more modular way
- Format:

```
return_type FunctionName( parameter1 , parameter2,... )
{
    //Statements
}
```

- Usage:

```
x = FunctionName( p1, p2, .. );
```

Warlocks 1507

# Functions

Example #1

```
#include <iostream>

//Function Prototypes
int Addition( int a, int b );

int main()
{
    int sum = Addition( 2, 3 );

    std::cout << "Sum is: " << sum << std::endl;
    return 0;
}


//Function Addition
int Addition( int a, int b )
{
    int c;
    c = a + b;
    return(c);
}
```

# In Class Example Program

In class example: **Calculate factorial**

Known:

- n! = n * (n-1) * (n-2) * … * 1
- 0! = 1
- n must be positive whole number
- n < 13 (because 13! > 2^31 (int) )

# Assignment #3: Guessing Game

Your c++ code will randomly choose a whole integer between 0 and 100 and the user must guess what it is. Your program will keep asking for a new number until the user correctly guesses the answer.

Your program must:

- Ask for a guess
- Check for valid inputs ( 0 ≤ guess ≤ 100 )

Must be a function

- If guess is too high, print “Too High!”
- If guess is too low, print “Too Low”
- If guess is correct:
  - Exit loop
  - output “Winner!”
  - output number of guesses it took

# Homework Problem Hints

- Hint #1: Random Number

```
#include <cstdlib>
#include <ctime>

//Generate Random Number
srand(time(NULL));
int random_number = rand() % 100;
```

- Hint #2: Barebones program
  - A barebones example program to help get started will be included in the class github repo

- Hint #3: Use std::cout to debug what your program is doing!