

## Control Structures

**Control structures** are statements that control how other commands are executed. Some of the control structures we will be looking at in this class are **repeat**, **while**, **do/while**, and **if/else**.

**1) repeat** – The **repeat** structure represents a loop; it causes a set of commands to be executed a certain number of times.

**Setup:**

```
repeat (5)           // The number of times goes in the parenthesis  
{ .....; }         // Looped commands are placed in brackets.
```

**2) while** – The **while** structure causes a set of commands to be executed as long as a particular condition is true. Once the condition becomes untrue, the loop is exited and the next part of the program begins to run.

**Setup:**

```
while(condition)   // Conditions are discussed below  
{ .....; }         // Looped commands are placed in brackets.
```

**Special Conditions:**

There are two special conditions that can be used with the while structure. They are i) **true**, which is a condition that is always true, and ii) **false**, which is a condition that is never true.

Commands inside a **while(true)** structure will be looped infinitely. Because the condition “true” is always true, the commands will always be executed.

Commands inside a **while(false)** structure will never run. Because the condition “false” is never true, the commands will never be executed.

**Numerical Conditions:**

Other while conditions can be made with the following comparison symbols:

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

### Example:

```
int VARIABLE = 0;
task main()
{
while(VARIABLE < 20)
{OnFwd(OUT_A + OUT_C);      Wait(100);
 Off(OUT_A + OUT_C);       Wait(100);
 VARIABLE +=5;}}
```

While the variable **VARIABLE** is less than 20, Motors A and C will move forward for 1 second, and then turn off for 1 second. **VARIABLE** increases by 5 at the end of each loop. At the end of the first loop, **VARIABLE** equals 5. At the end of the second loop, **VARIABLE** equals 10, third loop, 15, fourth loop, 20. After the fourth loop, the condition (**VARIABLE < 20**) is no longer true. Therefore, the commands are not executed again.

### Combining Conditions:

A loop can also be executed based on the truth value of more than one condition. These conditions can be brought together by the conjunctions and and or. Their symbolic representations are given below:

Symbol	Meaning
&&	And
	Or

**3) do/while** – The **do/while** structure is almost identical to the **while** structure, except that the condition is checked at the end of the structure. Therefore, the commands are executed at least once, even if the condition is false.

### Setup:

```
do
{.....;}
while(condition);
```

**4) if/else** – The **if/else** structure represents a fork. If a set condition is true, one set of commands is executed. If the condition is false, another set of commands is executed (the **else** part).

### Setup:

```
if (condition)
{.....;}
else
{.....;}
```