

Syntax

March 2021 J. Smallcombe

Syntax (c++ crash course)

Declaring a new local variable:

- All data stored in memory must have a type so the computer knows what it is and what to store
- Types are called “class”
- Simplest example integer number

int Identifier;

Syntax (c++ crash course)

Declaring a new local variable:

- All data stored in memory must have a type so the computer knows what it is and what to store
- Types are called “class”
- Simplest example integer number

int Identifier;



Class name - Tells computer what type to create in memory

Syntax (c++ crash course)

Declaring a new local variable:

- All data stored in memory must have a type so the computer knows what it is and what to store
- Types are called “class”
- Simplest example integer number

int Identifier;



Identifier - Arbitrary characters we choose to identify this instance of the int class

Syntax (c++ crash course)

Declaring a new local variable:

- All data stored in memory must have a type so the computer knows what it is and what to store
- Types are called “class”
- Simplest example integer number

int Identifier;

Assigns the required amount of local memory needed to store an integer and assigns the default value.

Syntax (c++ crash course)

Declaring a new local variable:

- All data stored in memory must have a type so the computer knows what it is and what to store
- Types are called “class”
- Simplest example integer number

int Identifier;

Local memory means the memory will be freed (no longer assigned to the specified variables) at the end of the current environmental scope (current function/loop/etc).

Syntax (c++ crash course)

Declaring a new local variable, using constructor:

- All classes in c++ have a constructor function that requires no arguments

int Identifier;

Same as

int Identifier();

- Most have additional constructors that take arguments so that the variable is created with non-default values

int Identifier(4);

Syntax (c++ crash course)

Declaration and assignment:

- Most classes support variable assignment with the **equals** operator.

```
int Identifier;  
Identifier=4;
```

- This can be done at the same time as declaring the variable

```
int Identifier=4;
```

Same as

```
int Identifier(4);
```


Syntax (c++ crash course)

Pointers:

- A pointer (indicated by an **asterisk** on the data type) holds the memory address of a variable. It is a variable itself, but pointers to all classes are the same, they are not the class they point to, they are only a memory address.

```
int* MyPointer;
```

The **ampersand** operator can be used to return the address of a variable, which can be assigned to a pointer

```
int Identifier;
```

```
MyPointer=&Identifier;
```

Syntax (c++ crash course)

Pointers and heap:

The Heap is shared **non-local** memory, it is never cleared automatically and variables on the heap exist until specifically deleted.

The key word **new** before a class name specifies to create a instance of the class on the heap.

new int(4);

Heap variables do not have identifiers, as they are not local, the constructor inputs instead appear directly after the class name.

Syntax (c++ crash course)

Pointers and heap:

When key word **new** is used a pointer to the heap variable is returned, which can be assigned to a local pointer-type variable in order to access the heap variable further (if needed)

```
int* pointer = new int(4);
```

Syntax (c++ crash course)

Class functions

For complex classes containing multiple datum, useful functions can be predefined as part of the class.

Functions are accessed by a dot following the identifier of the variable instance.

TVector3 Identifier;

(a geometric vector storing 3 Cartesian coordinates)

Some functions require input arguments

Identifier.SetZ(1.2);

Some functions return data

TVector3 Other = Identifier.Unit();

And some functions are neither

Identifier.Print();

Pointers and Class Functions

If we have a pointer we can access the functions of the class instance the pointer is pointing to using the **arrow** operator (The pointer has no functions, it is just a memory address NOT the class it points to)

```
TVector3* Pointer = new TVector3();
```

```
Pointer->SetZ(1.2);
```

```
TVector3 Other = Pointer->Unit();
```

```
Pointer->Print();
```

<http://www.cplusplus.com/doc/tutorial/arrays/>