## Basics

### initialization/declaration

traditional

int x = 20;

int y;

uniform initialization

int x {20};

int y;

### OOPs

object

properties

behaviour



## class

by default attributes/functions are private

in a struct default is public

aggregate initialization

when all members of a class are public

## constructor

* special member function
* automatically called when object of class is created
* initialise data
* if any setups required for class/object to be used
* same name as of the class
* no return type

### default constructor

* no parameters
* or has parameters that have all default values
* no user provided initialization was provided

## functions

* reusable code
* return values
  + if no return value, then write void

ODR

### function overloading

based on num of parameters

or type of parameters

never based on return values

avoiding ambiguous

explicitly typecasting

mixing overloaded 7 default values

default values

to the right most

no named arguments

## polymorphism

operator overload

C++ python

function overload

C++ java

## best practices

### declaration

uniform initialization

### functions

write return (even if it is void)

use const references wherever applicable

mention parameter names in function declarations

ODR

* a function/variable/type/template can have only one definition within a file

### class

class names with a capital letter

use struct for data-only structures

make member variables private, member functions public

(unless you have a good reason not to)

### strings

use std::string wherever possible

Herbert Schildt

The complete reference of C++

Scott Meyers

Effective Modern C++