Cheatsheets / **Learn C++**

codecademy

Classes & Objects

Destructor

For a C++ class, a *destructor* is a special method that handles object destruction, generally focused on preventing memory leaks. Class destructors don't take arguments as input and their names are always preceded by a tilde `~`.

```
City::~City() {

  // Any final cleanup

}
```

Destructor called automatically when obj. moves ou

Class Members

A class is comprised of class members:

- *Attributes*, also known as member data, consist of information about an instance of the class.
- *Methods*, also known as member functions, are functions that can be used with an instance of the class.

```
class City {

  // Attribute
  int population;

public:
  // Method
  void add_resident() {
    population++;
  }

};
```

Notice this semicolon!!!

Constructor

For a C++ class, a *constructor* is a special kind of method that enables control regarding how the objects of a class should be created. Different class constructors can be specified for the same class, but each constructor signature must be unique.

```
~~#include "city.hpp"~~

class City {

  std::string name;
  int population;

public:
  City(std::string new_name, int new_pop);

};
```

Then define constructor in .cpp file. You can do this in 2 w

Objects

In C++, an *object* is an instance of a class that encapsulates data and functionality pertaining to that data.

```
City nyc;
```

Access Control Operators

C++ classes have access control operators that designate the scope of class members:

- `public`
- `private`

`public` members are accessible everywhere; `private` members can only be accessed from within the same instance of the class or from friends classes.

by default everything is private.Use "public:" to decla

```
class City {

  int population;

public:
  void add_resident() {
    population++;
  }

private:
  bool is_capital;

};
```

Class

A C++ class is a user-defined data type that encapsulates information and behavior about an object. It serves as a blueprint for future inherited classes.

```
class Person {

};
```

You can also define class methods outside of the class using the syntax "Class::method"You can define cla