Top Down Design

- Break down a program/project to gain insight into the functions that make it up
- First, we create the overview of the program (main)
  - we specify but don't define the inner workings of the functions/steps
- Second, we define the functions/steps – giving more information
- Third, we add the rest of the code
- I am going to show you how to do this with an example in a few slides
- The book shows a method using a structure chart

Functional Programming VS Object Oriented Programming

- Object Oriented programming – C++
  - Develop classes – an object is an instance of a class
    - Contain Data
    - Run procedures known as methods
    - Can change state/mode
- Functional programming - C
  - Develop functions
    - computations
      - expressions
    - Do one small part of a bigger job
  - Logic
  - Doesn’t change state

Functional Programming VS Object Oriented Programming

- Difference is how you approach the problem you are trying to solve
- In all programs, there are two primary components: the data (the stuff a program knows) and the behaviors (the stuff a program can do to/with that data). OOP says that bringing together data and its associated behavior in a single location (called an “object”) makes it easier to understand how a program works. FP says that data and behavior are distinctively different things and should be kept separate for clarity.

From https://www.codenewbie.org/blogs/object-oriented-programming-vs-functional-programming

Functional Programming VS Object Oriented Programming

- Example: Give everyone in your organization a raise
- OOP:
  - Class for an employee, 3 attributes – name, id, salary
  - Now we create an object for each employee (pulling the info from a database)
  - Use a method to increase his salary
  - Use a method to update the database for that object
- FP:
  - Loop until end of data
  - Create a function to read one set of data from the database
  - This data includes the name, id and salary of the employee
  - Create another function to that increases the salary
  - Create another function to update that data element into the database

From https://www.codenewbie.org/blogs/object-oriented-programming-vs-functional-programming

Functional Programming VS Object Oriented Programming

- C vs C++ - same syntax for the most part
  - with additional syntax in C++ to allow for developing objects/classes
- Object-oriented programming (OOP) is a programming language model organized around objects rather than "actions" and data rather than logic. Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data
- objects we want to manipulate rather than the logic required to manipulate them.
- An example
  - a student
    - functional – use a data structure to store information about each student – name, email, grades, and we will determine his grade by creating logic or functions to add and multiply with percentages and come up with his grade and the class average
    - object oriented – we will create a class called student
      - each student will be represented by data – name, email, grade. When we manipulate the grade, we do this by working in the student object.

Why Learn C++

- C++ is known to be a very powerful language. C++ allows you to have a lot of control as to how you use computer resources, so in the right hands its speed and ability to cheaply use resources should be able to surpass other languages. Thanks to C++'s performance, it is often used to develop game engines, games, and desktop apps. Many AAA title video games are built with C++.

Taken from www.bestprogrammlanguagefor.me/why-learn-c-plus-plus

Object Oriented Programming

- Class
  - has attributes (data)
  - has behavior (functions – called methods)
    - special method – constructor – called whenever a new object is created
    - usually use set and get methods for accessing data
- Object
  - instance of a class

Classes

- think of it as a blueprint for creating an object
- in c terms – the class is the data type (struct) and the object is the variable of that type.
- In C the unit of programming is the function – data supports the actions that the functions are to perform
- In C++ the unit of programming is the class from which objects are instantiated
- think of it as nouns (C++) vs verbs (C – functions)

Comes down to how you design your program

Car Simulation Example

C

```
typedef struct {
        char* color;
        char* model;
        int id;
        char fuel_type;
} Car;

Car myford;
// Functions start, brake, accelerate
can be called from anywhere and you
must pass it the variable of type car
strcpy(myford.color, “blue”);
start(myford);
```

Car Simulation Example

C++

```
Class Car {
public:
        Car(string, string, long int, char);    // constructor
        void start();
        void accelerate();
        void brake();
private:
        string color;
        string model;
        long int id;
        char fuel_type;
}

Car myford("blue", …);    // will call constructor
…
myford.start();
```

C

```
typedef struct {
        char* color;
        char* model;
        int id;
        char fuel_type;
} Car;

Car myford;
// Functions start, brake, accelerate can be called from anywhere
and you must pass it the variable of type car
strcpy(myford.color, "blue");
…
start(myford);
```