Download

Day 9: Static

Download

Agenda

- Object Oriented Programming
- Static Methods
- Lab 5: Tic Tac Toe

Download

Object Oriented Programming

What is it?

- A different way of programming!
- Allows for data to be stored and "encapsulated"
- Think of it as your own custom data type that contains data inside of it!

Download

Why

Object Oriented Programming, or OOP for short, allows us to create structures that represent data in a way that allows us to manipulate it in a contained and consistent manner.

Download

Classes and Objects

Download

In OOP, there are two major topics in the form of "classes" and "objects"

Class:

The structure, definition of data, and how the data is manipulated.

Contains "methods"

Defines the variables and data stored inside.

Object:

The actual "version" or copy or instance of the class.

How we call those "methods" and functions.

Allows for multiple different "copies" of the class, following the structure.

Download

Blueprint Analogies

Download

A common analogy for Classes/Objects is a Blueprint/Actual.

The class is a blueprint to construct something. The object is the actual "thing" being constructed.

A Recipe/Dish is another good analogy. The Recipe for some dish is the class. It contains the structure, the ingredients, the order of how the dish is created. The Actual Dish created is the Object. You can follow the same instructions and create the same dish multiple times (creating multiple objects from the same class).

Download

We can make it a little more abstract too. Let's say a generic Computer is a class. The Object would be an individual computer, be it a Laptop or Desktop. We can have multiple computers that follow the same structure but with differences too. They are all computers and are "instances" or examples of a computer.

Download

4 Pillars

Download

4 Pillars

- Abstraction
- Encapsulation
- Polymorphism
- Inheritance

Download

Abstraction

The ability to take code that has similar functionality and move it into a more generalized implementation.

We won't see this too much in this class, but it'll show up more in 112 with the implementation of interfaces and abstract classes.

Download

Encapsulation

The ability to hide and store data within some class or object. This is heavily done using the `private` or `public` keywords.

Data is hidden, using methods to expose and manipulate the data.

We will see this next week when we start creating our own classes!

Download

Inheritance

The ability for classes to inherit data from parent classes using the `extends` and `super` keywords.

We will see this near the end of the semester when we properly talk about inheritance. This will also show up more in 112.

Download

Polymorphism

The ability to have different implementations for methods/functions, with different function parameters.

We will see this more next week too!

Download

Wait...

Now, that was more of an introduction to Object Oriented Programming from a more theoretical/abstract perspective.

We will properly work through creating our own classes next week

Download

...why?

Before we start implementing our own classes, we should look at how to create functions in our code in order to clean up our code.

Download

Today, we're going to dive into the `static` keyword.

Static Methods

Let's look at the main method again. Remember how I said we'll talk about that `static` keyword later?

Well, now is later.

```
public static void main(String[]
args) {

  System.out.println("Hello World");

}
```

Using the static keyword allows us to write functions that don't require us to work with classes and objects quite yet.

```
public static void main(String[]
args) {

  System.out.println("Hello World");

}
```

We can call static functions within other static functions. We can also call static functions in non-static methods.

However, we cannot call non-static methods in static functions.

```
public static void main(String[]
args) {

  System.out.println("Hello World");

}
```

Creating static functions

Creating static functions is really straightforward. We simply add the static keyword after the `public/private` and before our `return type`

```
public static <type> <name>(params)
{

  // function body

}
```

Creating static functions

We then give our function a name, parentheses, and any parameters inside of it.

This is followed by curly brackets.

Inside the function, we can do any calculation and return a value.

This value returned should match the return type of the function.

```
public static <type> <name>(params)
{

  // function body

}
```

Download

Creating static functions

In the example to the right, we have a static function that takes two integer parameters, `a` and `b`, with an integer return type. The function is named `add` and simply returns the sum of both numbers.

```
public static int add(int a, int b)
{

  return a + b;

}
```

Download

Creating static functions

In the case to the right, we don't return anything. This version simply prints out a value and doesn't **return** a value. In this case, we can change the return type into a `void` type. This version of our `add` function simply prints out the sum of `a` and `b`.

Void functions **do not** return any value! The `return` keyword can be used to end the function, but it cannot return an actual value.

```
public static void add(int a, int
b) {

  System.out.println(a + b);

}
```

It is important to always remember that we need a return type in our functions!

Not having one will cause the compiler to complain and cause our code to fail to compile.

Download

Count Lowercase Letters Function

```
public static int
countLowers(String str) {
  int total = 0;
  char[] chrs = str.toCharArray();
  for(char c : chrs) {
    if (c >= 'a' && c <= 'z') {
      total++;
    }
  }
  return total;
}
```

```
System.out.println(countLowers("HElLo"));
System.out.println(countLowers("world"));
```

Download

What's the point of writing static functions?

Download

- Allows us to write code that will get called multiple times; code reusability
- Code can be more generic and can work for multiple cases
- Allows us to avoid writing the same code in different places
- Essentially, for now, they are just functions that we can call in the main method!

Download

Interactive Grading: Project 1

Project 1 is due on Friday, 9/26. Monday, next week, 9/29 will go differently.

Plan:

- The lecture will be recorded and posted over the weekend. Check CampusWire for the announcement and the video link.
- We will spend the first 10 minutes in class for the Quiz. Come for that.
- After the quiz, we may go over it, but you will be allowed to leave. Come back for your Allocated Time slot for the Interactive Grading of Project 1.
- In the meantime, work on the in-class assigned.

Sign up for your 5 minute meeting with myself/your TA in the Google Spreadsheet.

The Interactive Grading will be a brief 5 minute conversation with either myself or your TA. We simply want to check your understanding of your project and the code you wrote.

Download

Lab 5: Tic Tac Toe