Java Programming Foundations 1

Week 9: Objects and classes

Overview

- Objects
- Classes
- Syntax
- Abstractions
- Encapsulation
- The `File` class
- Imports
- Types

Objects

What are objects?

Objects are a representation or an abstraction of an entity such as a car or a person.

Classes

What are classes?

Classes allow you to define data (attributes) and actions (methods) that relate to an entity.

How do classes relate to objects?

How do classes relate to objects?

Objects are created using classes.

How do classes relate to objects?

Objects are the instantiation of a class.

How do classes relate to objects?

Classes are the blueprints for objects.

You're already working with classes

You're already working with classes

Every time you create a new file in BlueJ, that's a class.

You're already working with classes

Java comes with can use built-in classes.

You're already working with classes

- `Math`
- `Scanner`
- `String`
- `System`

Syntax

Syntax

```
// Modifiers
//   |
//   |  The word "class"
//   |    |
//   |    |    Name of the class
//   |    |       |
//   |    |       |
//   V    V       V
public class Calculator
{
// <------------------------+
//                          |
//                          |
//                          +---- Class body
//                          |
//                          |
// <------------------------+
}
```

Sample of a class

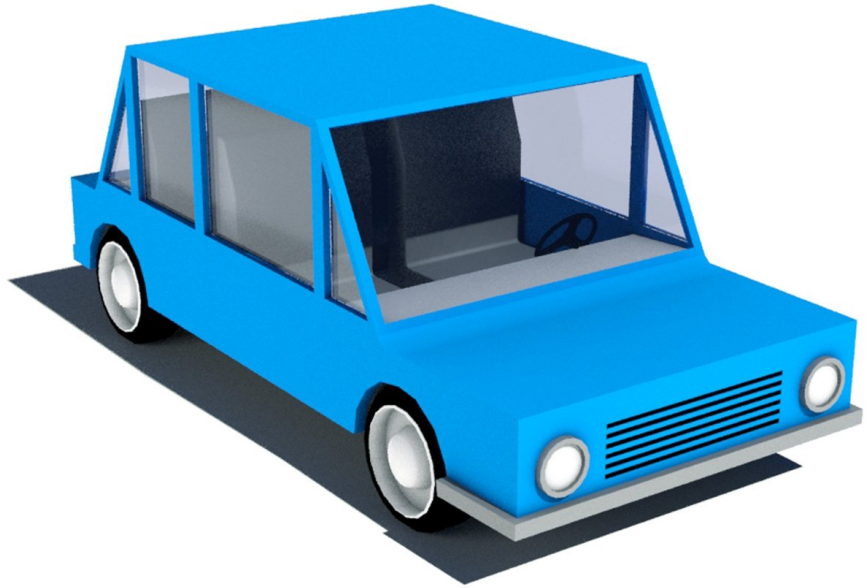
```
public class Calculator
{
    public static double add(double a, double b)
    {
        return a + b;
    }

    public static double subtract(double a, double b)
    {
        return a - b;
    }

    public static double multiply(double a, double b)
    {
        return a * b;
    }

    public static double divide(double a, double b)
    {
        return a / b;
    }
}
```

Abstractions

What is an abstraction?

An abstraction is the representation of an idea.

Let's define a sample class for a car



Let's define a sample class for a car

- What are the attributes a car has?
- What are the actions that a car can perform?

What are the attributes of a car?

- Exterior color
- Interior color
- Number of doors
- Number of wheels
- Maximum number of passengers

What are the actions a car can perform?

- Drive forward
- Drive backward
- Turn left
- Turn right
- Break

Let's write some code

The `Car` class

```
public class Car
{
    public static void driveForward()
    {
        System.out.println("Driving forwards");
    }

    public static void driveBackward()
    {
        System.out.println("Driving backwards");
    }

    public static void turnLeft()
    {
        System.out.println("Turning left");
    }

    public static void turnRight()
    {
        System.out.println("Turning right");
    }

    public static void stop()
    {
        System.out.println("Stopping");
    }
}
```

The `Runner` class

```
public class Runner
{
    public static void main(String[] args) throws IOException
    {
        Car.driveForward();
        Car.turnLeft();
        Car.turnRight();
        Car.stop();
    }
}
```

Encapsulation

What is encapsulation?

Encapsulation is the packaging or grouping of logic and behaviour into a unit.

What is encapsulation?

Encapsulation is the packaging or grouping of logic and behaviour into a class.

The packaging of behaviour into a class

Car related code goes in the `Car` class.

The `File` class

What can it do?

Allows you to read and write to files in your computer.

Let's write some code

Print the contents of a file using the `File` and `Scanner` classes.

```java
import java.io.*;
import java.util.*;

public class PrintFile
{
    public static void main(String[] args) throws IOException
    {
        System.out.print("Path to a file: ");
        Scanner userInput = new Scanner(System.in);
        String inputFilePath = userInput.nextLine();

        File inputFile = new File(inputFilePath);
        Scanner inputFileScanner = new Scanner(inputFile);

        while (inputFileScanner.hasNext()) {
            String line = inputFileScanner.nextLine();
            System.out.println(line);
        }
    }
}
```

Objects are the instantiation of a class

Instantiation is to create a "copy" of something.

Objects are the instantiation of a class

Use the `new` keyword to instantiate a class.

Objects are the instantiation of a class

```
File inputFile = new File(inputFilePath);
Scanner inputFileScanner = new Scanner(inputFile);
```

Imports

Imports

Use the `import` keyword to gain access to a class defined externally.

Imports

```
import java.io.*;
import java.util.*;
```

Imports

- `import java.io.*` gives you access to `File` and `IOException`,
- `import java.util.*` gives you access to `Scanner`

Types

Types

Every class is a type.

Every class is a type

```
Scanner userInput = new Scanner(System.in);
String inputFilePath = userInput.nextLine();

File inputFile = new File(inputFilePath);
Scanner inputFileScanner = new Scanner(inputFile);
```