codecademy

Classes

Classes

A class is an object-oriented concept which resembles a blueprint for individual objects. A class can contain properties and functions and is defined using the `class` keyword followed by a name and optional body. If the class does not have a body, its curly braces can be omitted.

```
// A class with properties that contain de
fault values
class Student {
  var name = "Lucia"
  var semester = "Fall"
  var gpa = 3.95
}

// Shorthand syntax with no class body
class Student
```

Class Instances

A new instance of a class is created by calling the class name followed by a pair of parentheses `()` and any necessary arguments.
When creating an instance of a class, we must declare a variable in which we intend to store our instance and assign it equal to the class call. Once the instance has been created, we can use dot syntax to access and retrieve the value of each property.

```
// Class
class Student {
  var name = "Lucia"
  var semester = "Fall"
  var gpa = 3.95
}

fun main() {
  var student = Student()   // Instance
  println(student.name)     // Prints: Luc
ia
  println(student.semester) // Prints: Fal
l
  println(student.gpa)      // Prints: 3.9
5
}
```

Primary Constructor

A primary constructor defines each property value within a class header and allows us to then set unique values when the object is instantiated.
To create a primary constructor using the shorthand syntax, we can follow the name of the class with a pair of parentheses `()` inside of which each property is defined and assigned a data type.

```
class Student(val name: String, val gpa: D
ouble, val semester: String, val estimated
GraduationYear: Int)

fun main() {
    var student = Student("Lucia", 3.95, "
Fall", 2022)
  println(student.name)     // Prints: Luc
ia
  println(student.gpa)      // Prints: 3.9
5
  println(student.semester) // Prints: Fal
l
  println(student.estimatedGraduationYear)
 // Prints: 2022
}
```

Init Blocks

The `init` block gets invoked with every instance that's created and is used to add logic to the class. The `init` keyword precedes any member functions and is followed by a pair of curly braces.

```
class Student(val name: String, val gpa: D
ouble, val semester: String, val estimated
GraduationYear: Int) {

  init {
    println("$name has ${estimatedGraduati
onYear - 2020} years left in college.")
  }
}

fun main() {
  var student = Student("Lucia", 3.95, "Fa
ll", 2022) // Prints: Lucia has 2 years le
ft in college.
}
```

Member Functions

A function declared within a class is known as a member function of that class. In order to invoke a member function, we must call the function on an instance of the class.

```
class Student(val name: String, val gpa: Double, val semester: String, val estimatedGraduationYear: Int) {

  init {
    println("$name has ${estimatedGraduationYear - 2020} years left in college.")
  }

  // Member Function
  fun calculateLetterGrade(): String {
    return when {
      gpa >= 3.0 -> "A"
      gpa >= 2.7 -> "B"
      gpa >= 1.7 -> "C"
      gpa >= 1.0 -> "D"
      else -> "E"
    }
  }
}

// When an instance is created and the function is called, the when expression will execute and return a letter grade

fun main() {
  var student = Student("Lucia", 3.95, "Fall", 2022) // Prints: Lucia has 2 years left in college.
  println("${student.name}'s letter grade is ${student.calculateLetterGrade()}.") // Prints: Lucia's letter grade is A.
}
```