Introduction to Java

LESSON 1

Learning Outcomes: Lesson 1

Classes

Objects

Classes vs Objects

Instance Variables

Data type

Methods

Naming conventions

Visibility Modifiers

Method : return type and parameter

Null

Constructor

Accessors

Mutators

Class

- A Java class is a file
- The file describes a general category in terms of:
  (a) data and
  (b) behaviors
- For example: we can write a Book class, which describes all books. All books have a title, an author, a date written.
- Another example: we can write a BankAccount class, which describes all bank accounts.
- All bank accounts have an account number, a date opened, and a balance (say, in US dollars).

Object

- Whereas a class describes a general category, an object is one specific instance of a class.
- For example, once you have described what a Book is, and that it has a title, an author, and a date written, you can then produce Book objects.
  - Harry Potter is the title of a Book written in 1995 by J.K. Rowling
  - Lord of the Rings is the title of a Book written in 1949 by J.R.R. Tolkien

Class vs Object

- Class is the template for the object. It works like a blueprint from which object is created. While object is the instance of the class.
- Class is the logical entity while Object is the physical entity.
- Class is declared once while object can created as many as requirement.
- Class doesnot allocated memory when it is created while object allocates memory when it is created.

Instance Variable

- Each piece of data in a class is known as an instance variable.
- It may be also called a field, a data member, or other names.
- Each instance variable has a data type.
- For example, the title of a Book is a String.
- The year a Book is written is an int.
- The balance in U.S. dollars of a bank account is a double.

Data Type

Java has two categories of data types.

- Primitive data types
  - int
  - double
  - boolean
  - char
- Reference (or Object) types
  - String
  - Book and BankAccount are also reference types. Classes are reference types.
  - Note the Capital first letters of these data types

Methods

- In addition to instance variables, classes contain behaviors
- Behaviors are called methods inside a class.
- BankAccount methods could include
  - withdraw()
  - deposit()
  - transfer()
  - getBalanceUSD()
  - changePIN()

Naming Conventions

- Class names should be nouns, in mixed case with the first letter of each internal word capitalized. For example BookAccount or Book
- Methods should be verbs, in mixed case with the first letter lowercase and with the first letter of each internal word capitalized. For example transfer() or changePin()
- Variable names should be short yet meaningful. One-character variable names should be avoided except for temporary variables.
  - Constant variable Should be all uppercase with words separated by underscores (“_”).

Visibility Modifiers

- Visibility modifiers are also called access modifiers.
- Instance variables, methods, and constructors can be declared as public or private (there are also other modifiers, to be discussed later).
- We will always make our instance variables private.
- We will make our classes and constructors public.
- We will make some methods public, others private.
- Private means **accessible only from within this class**.
- Public means **accessible from any class**.

Null

- Null means no object (no reference) for object / reference data types.
- Not relevant for primitive data types.

Methods : Return type and Parameters

- Java methods can take input (parameters) and give output (return).
- Inputs are called parameters or arguments.
- If a particular method doesn’t need to return anything, its return type is void.
- A method must return data of the specified return type, or it will not even compile.
- A method signature includes:
  - modifier return-type name(parameters)

Constructor

A constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created.

- A constructor in a class has the exact same name as the class.
- It is called automatically when an object is created.
- Its job is to ensure that the instance variables are set to proper/normal/valid/sensible initial values.
- A constructor should not be marked private.

Accessors

- Accessors methods are known as **get methods** or **getter**
- Accessor methods give other classes access to private data (instance variables, for example).
- The common method signature for accessor methods:
  - public <return type> getInstanceVariableName()
- Usually, no parameters are necessary.
- The return type matches the type of the instance variable being returned.
- Often, the accessor method does nothing except simply return data.

Mutators

- Mutators are also known as **set method** or **setter.**
- They allows other classes to change instance variables.
- The common method signature for mutator methods:
  - public void setInstanceVariableName(<data type> newVariableValue)
- Mutators generally have rules that must be followed first; if the rules are not followed, then the data is not changed.
- These rules are what make mutators useful. We protect the data this way.
- Without any guard rules, we may as well have made the instance variables public (a bad idea since any code anywhere could change the data to any value).