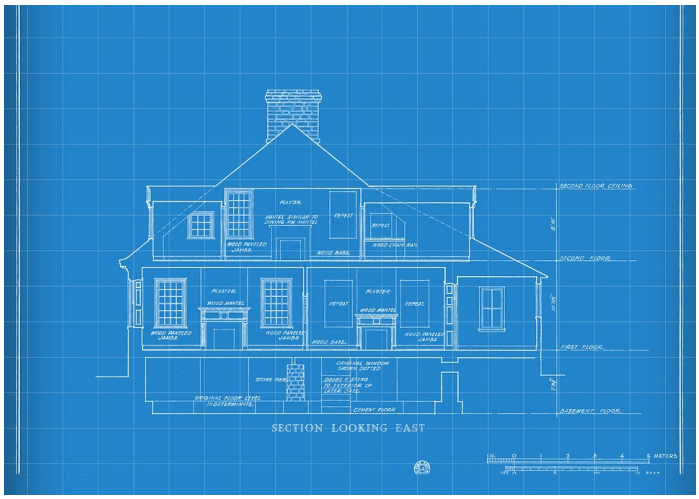
A Brief Introduction to Objects and Classes

In the following lesson, you will be given a very brief introduction to objects and classes.

We'll cover the following

Classes are Blueprints

A blueprint is a guide used for making something. In the real world, blueprints are usually plans for making buildings. You can create multiple buildings using the same blueprint with each building being unique but having the same basic architecture. For instance, a blueprint might specify the number of rooms a house should have. While each house built using that blueprint will have the same number of rooms, one house might have rooms with white walls and another might have rooms with blue walls, making them unique entities, but still related through the blueprint.



Classes can be thought of as a blueprint with which you can create an **object**. An object is an **instance** of a class which is the actual thing to be used in our programs. In the same way, a single house is an instance of the blueprint above. The blueprint is just a piece of paper while the house itself is the physical representation of that blueprint and can actually be lived in.

The blueprint will specify the properties the object would have and will also specify the operations/methods which the object can use. The properties and methods of a class are known as the **members** of that class.

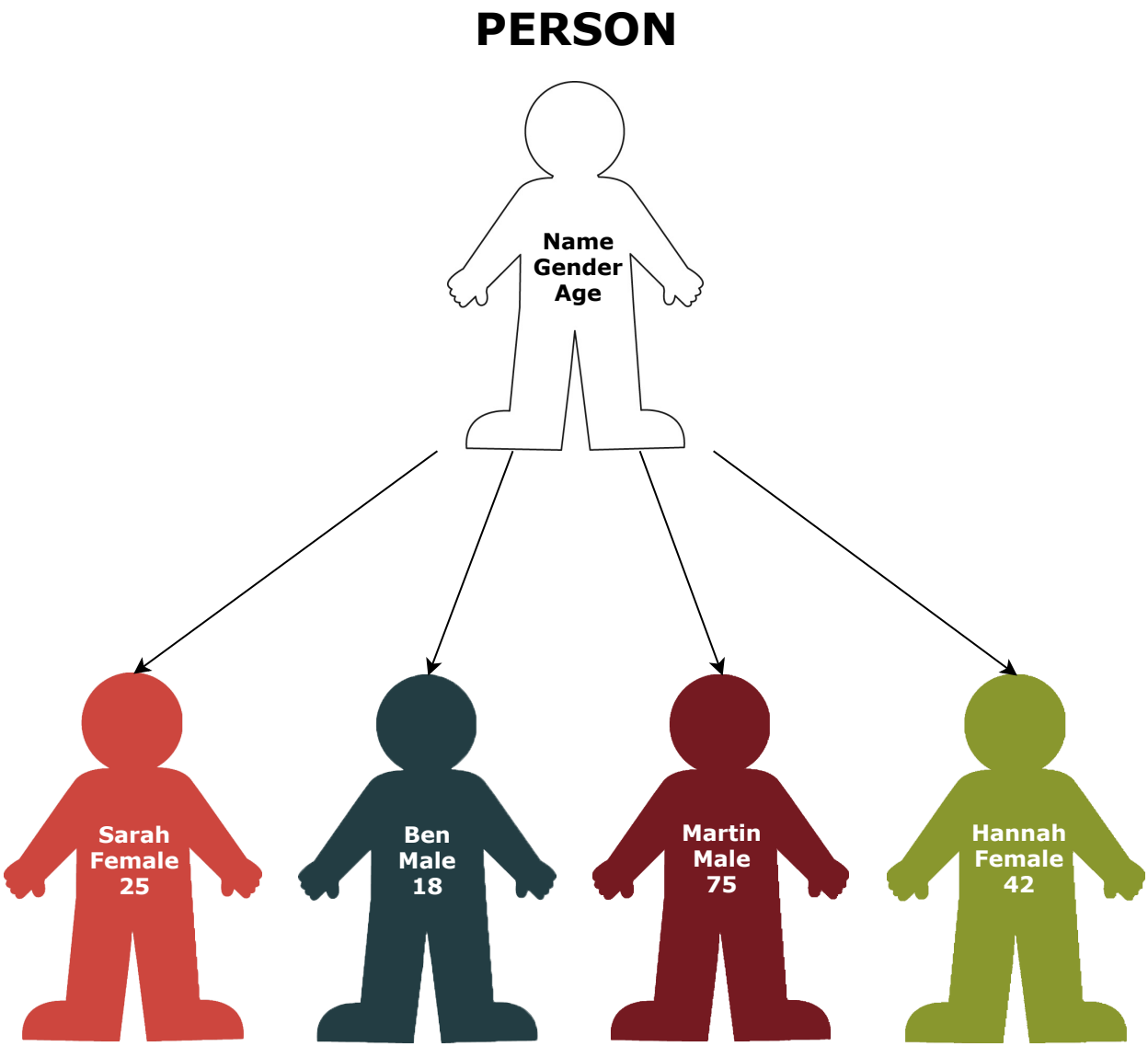
As always, learning by example is the best way to learn, so let's imagine how we can make a class which can be used to create a person.

Learning by Example

To make our person class, we first need to define its members, i.e., properties and methods.

Every person has a **name**, a **gender**, and an **age**. These are the properties of our person class. Furthermore, a person can perform certain actions such as **walking** and **talking**. These are the methods of our function class.

When we use our person class to create a *person*, we will create an instance of a person with its unique properties. We can create multiple instances using the same class.



Each person, Sarah, Ben, Martin, and Hannah can perform the methods of *walking* and *talking*.

Built-In Classes

Classes in Scala are broadly divided into two categories; built-in classes and user-defined classes. The focus of this chapter will be built-in classes, i.e., classes provided by Scala.

Do you remember when we discussed strings? All the methods we discussed were actually a part of Scala's built-in `String` class. This is how strings differ from other data types. When we define a string, we are actually creating an instance/object of the `String` class. Methods such as `length` and `toUpperCase` are all members of the `String` class which can be used by an object of that class.

In the upcoming lessons, we will be looking at Scala's collection library, which is made up of multiple classes.

In the next lesson, you will be introduced to the Scala collection library.