

R OOP (Objects and Classes Introduction)



R OOP (Objects and Classes Introduction)

R has 3 classes. In this article, you'll be introduced to all three classes (S3, S4 and reference class) in R programming.

We can do object oriented programming in R. In fact, everything in R is an object.

An object is a data structure having some attributes and methods which act on its attributes.

Class is a blueprint for the object. We can think of class like a sketch (prototype) of a house. It contains all the details about the floors, doors, windows etc. Based on these descriptions we build the house.

House is the object. As, many houses can be made from a description, we can create many objects from a class. An object is also called an instance of a class and the process of creating this object is called instantiation.

While most programming languages have a single class system, R has three class systems. Namely, S3, S4 and more recently Reference class systems.

They have their own features and peculiarities and choosing one over the other is a matter of preference. Below, we give a brief introduction to them.

S3 Class

S3 class is somewhat primitive in nature. It lacks a formal definition and object of this class can be created simply by adding a class attribute to it.

This simplicity accounts for the fact that it is widely used in R programming language. In fact most of the R built-in classes are of this type.

See R programming S3 Class section for further details.

R OOP (Objects and Classes Introduction)

Example 1: S3 class

```
> # create a list with required components
> s <- list(name = "John", age = 21, GPA = 3.5)
> # name the class appropriately
> class(s) <- "student"
```

Above example creates a S3 class with the given list.

S4 Class

S4 class are an improvement over the S3 class. They have a formally defined structure which helps in making object of the same class look more or less similar.

Class components are properly defined using the `setClass()` function and objects are created using the `new()` function.

See R programming S4 Class section for further details.

Example 2: S4 class

```
< setClass("student", slots=list(name="character", age="numeric", GPA="numeric"))
```

Reference Class

Reference class were introduced later, compared to the other two. It is more similar to the object oriented programming we are used to seeing in other major programming languages.

Reference classes are basically S4 classed with an environment added to it.

See R programming Reference Class section for further details.

R OOP (Objects and Classes Introduction)

Example 3: Reference class

```
< setRefClass("student")
```

Comparison between S3 vs S4 vs Reference Class

S3 Class	S4 Class	Referene Class
Lacks formal definition	Class defined using <code>setClass()</code>	Class defined using <code>setRefClass()</code>
Objects are created by setting the class attribute	Objects are created using <code>new()</code>	Objects are created using generator functions
Attributes are accessed using <code>\$</code>	Attributes are accessed using <code>@</code>	Attributes are accessed using <code>\$</code>
Methods belong to generic function	Methods belong to generic function	Methods belong to the class
Follows copy-on-modify semantics	Follows copy-on-modify semantics	Does not follow copy-on-modify semantics