**Module # 7 R Object: S3 vs. S4 assignment**

**1.**    **How do you tell what OO system (S3 vs. S4) an object is associated with?**

       These techniques can be used to ascertain whether an object is related to S3 or S4:

       To determine whether an object is not an S4 object, use is.object(x) &!isS4(x).

       To obtain the object type, use sloop::otype(x). It is connected to S3 if it yields "S3".

       While base objects lack a class attribute, S3 objects do.

**2.**      **How do you determine the base type (like integer or list) of an object?**

      To find an object's base type, use typeof(x). As an illustration,

      typeof(1:10) yields "integer".

      mtcars.typeof() yields "list".

**3    What is a generic function?**

    A generic function is a function that establishes a shared interface for functions that are  related. It enables you to call methods according to the input object's class.

S3 generic functions are non-formal and determine which method to call by means of a unique kind of function known as a generic function.

Generic functions in S4 support multiple dispatch and have formal class definitions.

**4.**  **What are the main differences between S3 and S4?**

**S3:**

Informal system.

No formal class definitions.

Methods belong to generic functions.

Objects can belong to multiple classes.

**S4:**

Formal system.

Has formal class definitions.

Methods are explicitly defined.

Objects belong to a single class.

**5.**  **Examples of S3 and S4:**

     Here are a few condensed examples:

**S3:**

# Create an S3 object

s <- list(name = "Arusmita", age = 24, GPA = 3.5)

class(s) <- "student"

**S4:**

# Create an S4 class definition

setClass("student", slots = c(name = "character", age = "numeric", GPA = "numeric"))

# Create an S4 object

s4 <- new("student", name = "Arusmita", age = 24, GPA = 3.5)