Object Oriented Programming

Contents

[**Object Oriented Programming** 3](#_Toc121572360)

[**Understanding Classes, Objects, Properties, and Methods** 3](#_Toc121572361)

[Returning an object 4](#_Toc121572362)

[Examples of Objects with Close Method 5](#_Toc121572363)

[Examples of Object in a Hierarchy with Close Method 5](#_Toc121572364)

[Enumeration 6](#_Toc121572365)

[There are eight types of enumeration: 6](#_Toc121572366)

[Closing 6](#_Toc121572367)

[Instantiation / Declaration of an Object 6](#_Toc121572368)

[Instantiation / Declaration of an Object 6](#_Toc121572369)

[Executable File and Languages 7](#_Toc121572370)

[Scripted Files and Languages 7](#_Toc121572371)

[Java and Scripting Languages 7](#_Toc121572372)

[Instantiation in Python 8](#_Toc121572373)

[instantiation in Java 8](#_Toc121572374)

[Instantiation in C# 8](#_Toc121572375)

[References 8](#_Toc121572376)

**Object Oriented Programming**

**Understanding Classes, Objects, Properties, and Methods**

* **Class** – Collection and Classification of Objects
* **Objects** – elements, parts, components that have properties and methods pertaining to that object
  + Document collection object contains subordinate objects
    - Text
    - Tables
    - Bookmarks
    - References
    - Pages
* **Property** – attributes / characteristics / qualifications / variables of an object - variables, flags, dimensions, periodicity, geography, features. Properties enable the holding and returning of information from an object. Objects can also store the state change for continued use. (A saved document will not become unsaved). This is sometimes called storing the quality of the object.

In a larger sense all properties are variables that are used to store values in memory. Variables have data types – character, number, constant. Variables describe the data that define the object

* + Document
    - Content
    - Content management
    - document form
    - Status – opened, closed, saved, unsaved, updated
    - Change tracking – on or off / true or false / yes or no
    - Date and time stamp
  + Employee
    - Name
    - Job title
    - Role
    - Responsiblities
    - Pay Rate
    - Vacation Time
    - Sick Time
    - Birthday
    - Email address
    - User Id and password
    - Payroll data
    - Work data
* **Methods** - actions that an object can perform. Methods cause an object to change (change of state)
  + Document
    - Create a new document
    - Open a current document
    - Print a document
    - Change the
    - Spell check a document
    - Change the size of text, paragraph, page, etc.
    - Change the format of text, paragraph, page, etc.
    - Save a document
    - Copy a document
    - Close a document
    - Change the tracking from on to off
    - File a document
    - Manage a collection of documents
    - Manage a collection of objects – documents, spreadsheets, presentations, etc.
  + Employee
    - Add Employee data
    - Change Employee data
    - Delete Employee data
    - Pay Employee
    - Pay Employee Taxes
    - Add Employee Benefits
* **Object Model** – All the object collections (classes) for a particular application. Also called the Application Model or the Product Model.

# Returning an object

Programs typically work on one occurrence of an object at a time. This occurrence is called an instance. At the top of a collection (or class) there is typically a root object that instantiates the next layer of objects. This sometimes referenced as the Application object or Application layer.

* Application Documents – Office Suite
  + Word
  + Excel
  + PowerPoints
* Application Employee
  + Payroll
  + Employee Reviews
  + Education and Training
  + Certifications
  + Benefits

When accessing (sometimes called instantiating) an object (typically a single object) from a collection, this single object referenced by an index number:

* Object\_name(1)
* Object\_name(index name)

## Examples of Objects with Close Method

**‘Close first document**

**Sub CloseDocument()**

**Documents(1).Close**

**End Sub**

**‘Close ("Sales.doc"document**

**Sub CloseSalesDoc()**

**Documents("Sales.doc").Close**

**End Sub**

## Examples of Object in a Hierarchy with Close Method

**Sub CloseDocSaveChanges()**

**ActiveDocument.Close SaveChanges:=wdSaveChanges**

**End Sub**

**‘Change Window State**

**Sub MaximizeDocumentWindow()**

**ActiveDocument.ActiveWindow.WindowState = wdWindowStateMaximize**

**End Sub**

The ActiveWindow property returns a Window object that represents the active window. The WindowState property is set to the maximize constant (wdWindowStateMaximize).

**‘Creates a document and displays the Save As dialog box so that a name can be provided for the document**

**Sub CreateSaveNewDocument()**

**Documents.Add.Save**

**End Sub**

The **Documents** property returns the **Documents** collection. The [**Add**](https://learn.microsoft.com/en-us/office/vba/api/word.documents.add) method creates a new document and returns a **Document** object. The **Save** method is then applied to the **Document** object.

## Enumeration

An enumeration has a name, an underlying data type, and a set of members. Each member represents a constant. An enumeration declared at class, structure, module, or interface level, outside any procedure, is a member enumeration. It is a member of the class, structure, module, or interface that declares it.

Enumerations make for clearer and more readable code, particularly when meaningful names are used. The benefits of using enumerations include: Reduces errors caused by transposing or mistyping numbers. Makes it easy to change values in the future.

An enumeration is a user-defined data type that consists of integral constants. To define an enumeration, keyword enum is used.

**enum season { spring, summer, autumn, winter };**

Here, the name of the enumeration is season .

**enum { Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday } weekday;**

defines the variable weekday , which can be assigned any of the specified enumeration constants.

### There are eight types of enumeration:

Windows enumeration, NetBIOS enumeration, LDAP enumeration, SNMP enumeration, Linux/UNIX enumeration, NTP enumeration, SMTP enumeration and DNS enumeration.

## Closing

you use methods or properties to drill down to an object. That is, you return an object by applying a method or property to an object above it in the object hierarchy. After you return the object that you want, you can apply the methods and control the properties of that object.

# Instantiation / Declaration of an Object

## Instantiation / Declaration of an Object

* Make an object active and ready for use (put it in memory for use)
* Giving the object a unique name that can be used in a program
* Giving a class a unique name that can be used in a program
* giving it a name and locating it in some physical place in memory
* An object is an instance of a class as defined by its specified execution attributes
* The object is an executable file that can run on a computer.
* Think of memory as the factory work floor. Objects are machines that are in inventory (typically disk) and are brought out of inventory to be used on the factory floor. This object on the floor is called an instance. The shop floor could have many instances on the floor being used in different applications, programs, or modules. Once they are finished being used, they are then removed from the factory floor.

## Executable File and Languages

* An executable file (EXE file) is a computer file that contains an encoded sequence of instructions that the system can execute directly when the user clicks the file icon. Executable files commonly have an EXE file extension, but there are hundreds of other executable file formats.
* Executable files contain binary machine code that has been compiled from source code. This low-level code instructs a computer's central processing unit on how to run a program. The processor interprets the machine code and tells the computer's hardware what to do.
* Structured and compiled languages - [C](https://www.techtarget.com/searchwindowsserver/definition/C) and [C++](https://searchsqlserver.techtarget.com/definition/C), Java

## Scripted Files and Languages

* Scripts are evaluated by a runtime environment (in this case, by a script engine) directly.
* Unlike source files for other programming languages that must be compiled into bytecode before you run them, scripts are evaluated by a runtime environment (in this case, by a script engine) directly.
* Most scripting languages are dynamically typed. This enables you to create new variables without declaring the variable type (the interpreter assigns the type based on the type of the object associated with the variable), and you can reuse the same variable for objects of different types (type conversion is performed automatically).
* Scripting languages generally have simple syntax; they allow complex tasks to be performed in relatively few steps.
* a script is a program or sequence of instructions that is interpreted or carried out by another program rather than by the computer processor (as a compiled program is).
* [Perl](https://www.techtarget.com/whatis/definition/Perl), Rexx (on IBM mainframes), [JavaScript](https://www.theserverside.com/definition/JavaScript), Python, PHP, Ruby, ASP.NET, Node.js, . In the context of the World Wide Web, Perl, [VBScript](https://www.techtarget.com/searchenterprisedesktop/definition/VBScript), and similar script languages are often written to handle forms input or other services for a Web site and are processed on the Web [server](https://www.techtarget.com/whatis/definition/server). A JavaScript script in a Web page runs "[client-side](https://www.techtarget.com/searchapparchitecture/definition/client-side-framework)" on the Web [browser](https://www.techtarget.com/whatis/definition/browser).

## Java and Scripting Languages

* Although scripting languages are usually interpreted at runtime, they can be compiled into Java bytecode that can then be executed on the Java Virtual Machine (JVM). Scripting languages can be faster and easier to use for certain problems, so it is sometimes chosen by developers of Java applications. However, if you write your Java application in a scripting language, then you lose the benefits of the Java language (such as type safety and access to the class library).
* Java Specification Request (JSR) 223: Scripting for the Java Platform addresses the issue of integrating Java and scripting languages. It defines a standard framework and application programming interface (API) to embed scripts in your Java applications and access Java objects from scripts.
* Features of Java
  + OOP programming model
  + Portable
  + Compilable – bytecode
  + Code is comprehensive
  + Is dynamic and flexible
  + Similar to C++
  + Runs in sandbox to provide security

## Instantiation in Python

* In Python, everything is an instance or an object.
* When a copy of a class is created that inherits all the class properties and functions, it is called instantiating a class.
* To instantiate a class in Python, the class like it is called a function, passing the arguments defined by the \_\_init\_\_ method. The newly created object is the return value.
* Advantages
  + Modular
  + Flexible due to dynamic typing, but also run-time errors
  + OOP
  + Extensive third-party libraries and modules
* Disadvantaged
  + Potentially complicated runtime environment setup
  + Slow execution speed
  + Lower memory efficiency

## instantiation in Java

* In Java, an OOP language, the object that is instantiated from a class is, confusingly enough, called a class instead of an object.
* In other words, using Java, a class is instantiated to create a specific class that is also an executable file that can run on a computer.
* Java's equivalent of a class attribute is a static attribute. Generally, static variables are referred to with the Java class name.

## Instantiation in C#

* In the OOP language C#, instantiation describes the processes of creating a new object for a class using a new keyword.

# References

* [Understanding Objects, Properties, and Methods](https://learn.microsoft.com/en-us/office/vba/word/concepts/objects-properties-methods/understanding-objects-properties-and-methods)
* [instantiation](https://www.techtarget.com/whatis/definition/instantiation)
* [DEFINITION: script](https://www.techtarget.com/whatis/definition/script)
* [Scripting Languages and Java](https://docs.oracle.com/javase/10/scripting/scripting-languages-and-java.htm#JSJSG107)
* [The Java Scripting API](https://docs.oracle.com/javase/10/scripting/java-scripting-api.htm#JSJSG109)
* [executable file (EXE file)](https://www.techtarget.com/whatis/definition/executable-file-exe-file)
* [C# (C-Sharp)](https://www.techtarget.com/whatis/definition/C-Sharp)