What is an Abstract Class in UML?

A **UML abstract class** is a class without any instances. It represents a common class for other derived subclasses. An abstract class can contain abstract and also non-abstract attributes and operations. A class that is not abstract cannot have any abstract fields (attributes, operations).

E.g. a class for Shape can be marked as abstract. It cannot be instantiated because we do not know what kind of shape it represents. It is a base (super) class for other shape classes (e.g. Ellipse, Rectangle).



UML abstract class vs. instantiable (non-abstract) class

Abstract Attribute

An **abstract attribute** is used to model an abstract property. The specific properties that can return an actual value are defined by subclasses using non-abstract attributes.

Abstract attributes are rendered as italic in UML.

Abstract Operation

An **abstract operation** models an **abstract method** in UML. It is also rendered in italic. The abstract operations do not have any logic defined. The concrete operations in the non-abstract subclasses define the behavior that is omitted in respective bodiless abstract operations.

Create a UML Abstract Class

You can set the **abstract modifier** for a [UML](https://www.softwareideas.net/uml) class in one of these ways:

1. Right-click on the class and choose Modifiers/Abstract from the context menu.

2. Open the Model tab in the Element context tab group and check the Abstract checkbox in the Modifiers group.

3. Double-click on the class, Properties dialog shows, check the Abstract checkbox in the Modifiers group, and confirm the choice using the OK button.

Create an Abstract Attribute or Operation

You can set the **abstract modifier** for an attribute or an operation in one of these ways:

1. Right-click on the attribute/operation and choose Modifiers/Abstract from the context menu.

2. Open the Model tab in the Field context tab group and check the Abstract checkbox in the Modifiers group.

3. Right-click on the attribute/operation and choose Properties from the context menu, Properties dialog shows, check the Abstract checkbox in the Modifiers group, and confirm the choice using the OK button.



**Abstract Classes**

It is a class with an operation prototype, but not the implementation. It is also possible to have an abstract class with no operations declared inside of it. An abstract is useful for identifying the functionalities across the classes. Let us consider an example of an abstract class. Suppose we have an abstract class called as a motion with a method or an operation declared inside of it. The method declared inside the abstract class is called a **move ()**.

This abstract class method can be used by any object such as a car, an animal, robot, etc. for changing the current position. It is efficient to use this abstract class method with an object because no implementation is provided for the given function. We can use it in any way for multiple objects.

In UML, the abstract class has the same notation as that of the class. The only difference between a class and an abstract class is that the class name is strictly written in an italic font.

An abstract class cannot be initialized or instantiated.

Abstract Class Notation

In the above abstract class notation, there is the only a single abstract method which can be used by multiple objects of classes.