

Difference Between Class and Instance Variables

www.differencebetween.com

Key Difference - Class vs Instance Variables

Most modern [programming languages](#) support [Object Oriented Programming](#). An [object](#) contains data or attributes. An object has certain behaviors. They are known as methods. A program can be model using objects. A [software](#) is a collection of programs. Therefore, a software can be designed and implemented using objects. Objects are interacted using methods. Object-Oriented Programming improves code modularity and reusability. There should be a class to create objects. A class is a blueprint to create an object. Therefore, an object is an instance of a class. In programming, the data needed to be stored. Data is stored in memory locations. These memory locations are called variables. A member variable is a variable that is associated with a specific object. It is accessible for all its methods. There are two types of member variables that are class variables and instance variables. The **key difference** between class and instance variables is that, **if there is only one copy of the variable shared with all instance of the class, those variables are called class variables and if each instance of the class has its own copy of the variable, then those variables are called instance variables.**

What are Class Variables?

When there is only one copy of the variable shared with all instance of the class, those variables are called as class variables. Class variables are variables declared within the class outside any method. These variables contain the keyword static. These variables are associated with the class, not to the object.

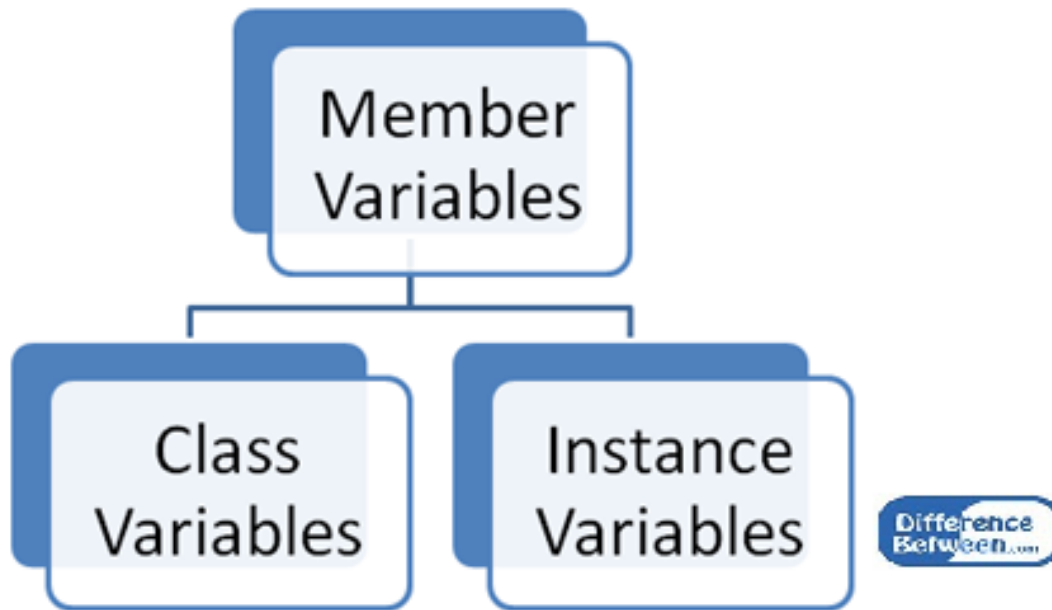


Figure 01: Class Variables and Instance Variables

Refer the below piece of code with class variables.

```
public class Employee {
public static int id;
public static double salary;

}

public class Test {
public static void main(string[] args){
Employee e1= new Employee();
Employee e2= new Employee();
}
}
```

According to the above program, e1 and e2 are Employee type objects. Both will have the same copy of memory. If e1.id= 1 and printing e2.id will also give the value 1. It is possible to print the id and salary values using the Employee class name such as Employee.id, Employee.salary etc.

What are Instance Variables?

When each instance of the class has its own copy of the variable, then those variables are known as instance variables. Refer the below program.

```
public class Employee {  
  
    public int id;  
  
    public double salary;  
  
}  
  
public class Test{  
  
    public static void main(string[] args){  
  
        Employee e1= new Employee();  
  
        e1.id=1;  
  
        e1.salary= 20000;  
  
        Employee e2= new Employee();  
  
        e2.id =2;  
  
        e2. salary = 25000;  
  
    }  
  
}
```

In the main program, e1 and e2 are references to the objects of type Employee. It is possible to assign values for id and salary using the dot operator such as e1.id, e1. salary etc. The id and salary in the class Employee are known as instance variables. The e1 and e2 are separate objects. Each object will have a separate

copy of instance variables. The e1 will have separate id and salary and e2 will have a separate id and salary. So, the instance variables are created when the object or the instance is created.

What are the Similarities Between Class and Instance Variables?

- Both are types of variables.
- Both variables are inside a class but outside any method.

What is the Difference Between Class and Instance Variables?

Class Variables vs Instance Variables	
Class variables are variables in which there is only one copy of the variable shared with all the instance of the class.	Instance variables are variables when each instance of the class has its own copy of the variable.
Association	
Class variables are associated with the class.	Instance variables are associated with objects.
Number of Copies	
Class variables create one copy for all objects.	Instance variables create separate copy for each object.
keywords	
Class variables should have the static keyword.	Instance variables do not require a special keyword such as static.

Summary - Class vs Instance Variables

Object-oriented programming is major programming paradigm. It helps to model a software using objects. Objects are created using classes. Object creation is also known as instantiation. A class provides a blueprint to create an object. A member variable is a variable that is associated with a specific object. It is accessible for all its methods. There are two types of member variables which are class variables and instance variables. The difference between class and instance variables is that, if there is only one copy of the variable shared with all instance of the class, those variables are called class variables and if each instance of the class has its own copy of the variable, then those variables are called instance variables.

Reference:

- 1.tutorialspoint.com. “Java Object and Classes.” [The Point. Available here](#)
- 2.“Instance variable.” Wikipedia, Wikimedia Foundation, 16 Dec. 2017. [Available here](#)
- 3.“Class variable.” Wikipedia, Wikimedia Foundation, 16 Dec. 2017. [Available here](#)

How to Cite this Article?

APA: Difference Between Class and Instance Variables. (2018 February 05). Retrieved (date), from <http://differencebetween.com/difference-between-class-and-vs-instance-variables/>

MLA: "Difference Between Class and Instance Variables". Difference Between.Com. 05 February 2018. Web.

Chicago: “Difference Between Class and Instance Variables”. Difference Between.Com. <http://differencebetween.com/difference-between-class-and-vs-instance-variables/>accessed (accessed [date]).



Copyright © 2010-2018 Difference Between. All rights reserved