SmallSEOTools

PLAGIARISM SCAN REPORT

Words	903	Date	May 21,2021
Characters	5748	Excluded URL	

7% Plagiarism	93% Unique	3 Plagiarized Sentences	42 Unique Sentences

Content Checked For Plagiarism

Blog on Different types of polymorphism and comparison of polymorphism in C++ and Java
Introduction
In the real world, you might have seen a chameleon changing its color as per its requirement. If someone asks, “How does it do so?”, you can simply say, “Because, it is polymorphic in nature”. Similarly, within the programming world, objects possess an equivalent functionality where each object can take multiple forms. This property is known as Polymorphism, where Poly means many and morph means change (or ‘form’). In this blog, let’s discuss this key concept of Object-Oriented Programming, i.e., Polymorphism in Java and C++.
Polymorphism in Object Oriented Programming is the ability of an entity to take several forms. In other words, it refers to the ability of an object (or a regard to an object) to take different forms of objects. It allows a standard data-gathering message to be sent to every class. Polymorphism encourages called as ‘extendibility’ which basically means an object or a class can have its uses extended.

In the above picture, you can see that the man is only one, but he takes multiple roles like -he is a dad to child, he is an employee, a salesperson and many more. This is known as Polymorphism.
Polymorphism in Java
Consider a mobile phone where you save your Contacts. Suppose a person has two contact numbers. For the ease, your cellphone provides you a function where you can save two numbers under the same name.
Similarly, in Java, an object is just one but it can take multiple forms counting on the context of the program. Let’s consider you want to create function to save two contact numbers of the same person, so you can create it like – void createContact(String name, int number1, int number2).
Now, it’s not necessary that everybody in your contact list will have two contact numbers. Few of them could be having only one contact number. In these kind of situations, instead of creating another method with a different name, what we can do is, create another method with the same name that is createContact(). But, instead of taking two or more contact numbers as parameters, we have to take only one contact number as a parameter that void createContact(String name, int number1).
createContact() method has two different definitions. Here, which definition is to be executed depends upon the no. of parameters being passed. If one parameter is passed, then only one contact number is saved under the contact. But, if two contact numbers are passed to the present method at an equivalent time, then both are going to be saved under an equivalent contact. This is also known as Method Overloading type of Polymorphism in Java
Below are two types of polymorphism in Java :
* Static Polymorphism
* Dynamic Polymorphism
Static Polymorphism
A polymorphism that's resolved during compile time is called as static polymorphism.
Method Overloading is a feature that permits a class to possess two or more method to possess an equivalent name, but with different parameter lists. In the below example, you've got two definitions of a similar method add(). So, which add() method would be called is decided by the parameter list at the compile time. That is the rationale this is often also referred to as compile time polymorphism.
Output:

Dynamic Polymorphism
Dynamic polymorphism is a process in which a call to an overridden method is resolved at runtime and because of this, we call it as a runtime polymorphism. Method Overriding is one of the ways to achieve Dynamic Polymorphism in java. Also, there is no operator overloading in Java. In any OOP language, Overriding is a feature which allows a child class to provide a specific implementation of a method(function) that is already provided by one of its super-classes or parent classes.
Output:
Polymorphism in C++
Basically in C++ polymorphism is divided into two types:
• Compile time Polymorphism
• Runtime Polymorphism
Compile time polymorphism: Compile time polymorphism can be achieved with the help of function overloading or operator overloading.
* Function Overloading: When there are more than one functions with same name but have different parameters then these functions are said to be overloaded. Functions are often overloaded by changing the number of arguments or/and in type of arguments.
Output:
* Operator Overloading: In C++ we can also achieve compile time polymorphism by overloading operators. For example, we can make use of the operator + for string class to concatenate two strings. We know that this is the addition operator which is used to add two operands. So this single operator '+' when we placed between integer operands it adds them and when we placed between string operands it concatenates them.
Output:
Runtime polymorphism: Runtime polymorphism is the type of polymorphism which is achieved with the help of Function Overriding.
* Function overriding : If the derived class defines same function as it was defined in its base class, then we can call this a function overriding. Function overriding is used to achieve runtime polymorphism.

Output:
Thus, in this blog we looked at the concept of Polymorphism through two different lenses, Java & C++. We looked at Static & Dynamic polymorphism, and later also looked at Runtime & Compile time polymorphism. It is fair to conclude from this that polymorphism is an incredibly useful part of a programmer's toolkit, which would help them to write clean & efficient code.

Sources	Similarity
Java for Selenium- Polymorphism (Method Overloading and ... · In the real world, you might have seen a chameleon changing its color as per its requirement. If someone asks, "How it does that?", you can simply say, "Because, it … https://www.youtube.com/watch?v=cbTEv1lZu_A	25%
characteristics of oops in java - Collectcent ... Contact. Simula is considered the first object-oriented programming language. ... In Java, we need to use the extends keyword to create a child class. ... S uppose you want to write a function to save two contact numbers of the same person, you can create it like – void createContact(String name, int number1, int number2). https://collectcent.com/old-admin/uploads/x/tpc-louisiana-vuvsv/3c9364-characteristics-of-oops-in-java	7%
C++ Function Overriding - javatpoint C++ Function Overriding. If derived class defines same function as defined in its base class, it is known as function overriding in C++. It is used to achieve runtime polymorphism. https://www.javatpoint.com/cpp-function-overriding	3%