What Do You Mean by Object Oriented Programming?

Author - Sakib Mahmud

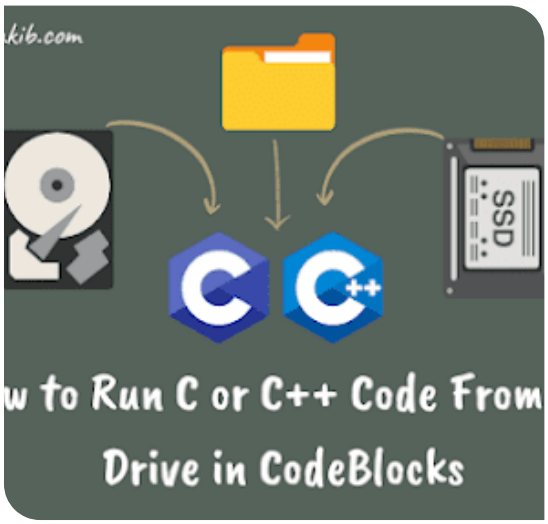
August 24, 2022 5 minute read 0

https://www.esakib.com/2022/08/object-oriented-programming.html Object Oriented Programming is a very important term for computer science students. We need to understand how this works and what this means. In this article, you will learn about object-oriented programming (OOP).

What is Object Oriented Programming (OOP)?

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields and code, in the form of procedures. Objects can be created from templates known as classes, which specify the kind of data and code that the object will contain. In most programming languages, objects are defined as classes, which specify the data and code the object will contain.

POPULAR ARTICLES



How to Run C or C++ Code From Any Drive in CodeBlocks

By - Sakib Mahmud in July 31, 2022

How to Run C/C++ in Visual Studio Code

August 01, 2022

What Do You Mean by Object Oriented Programming?

August 24, 2022

CodeBlocks Can't Find Compiler? Here Is the Solution!

August 14, 2022

Why Linux Is Better for Learning Programming

August 15, 2022

TOPICS

C C++ Linux-for-Programmers OOP Trending CS Topics Visual-Studio-Code

In object-oriented programming, an object is a self-contained unit of code and data. Objects are created from templates called classes. An object is an instance of a class. Classes are the fundamental units of code and data in object-oriented programming. A class is a template for an object. It is a blueprint that defines the variables and the methods common to all objects of a certain kind.

In object-oriented programming, computer programs are designed by making out of objects that interact with one another. There is significant diversity of OOP languages, but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

OOP is a way of thinking about programming that differs from the traditional procedural approach. OOP is a powerful programming paradigm that can help you design better, more reusable code. It can also make your code more maintainable and easier to understand. It is widely used in AI and ML.

Also read: AI vs Human Creativity

What are the characteristics of object-oriented programming?

There are four main characteristics of object-oriented programming:

1. Encapsulation
2. Abstraction
3. Inheritance
4. Polymorphism

Encapsulation

Encapsulation is the process of hiding data and code inside an object so that it can't be accessed

or changed by code outside the object. This helps keep the data and code safe from being accidentally, deliberately changed, or deleted.

Abstraction

Abstraction is the process of hiding the details of how an object works so that the code that uses the object doesn't need to know those details. This makes the code simpler and easier to understand and maintain.

Inheritance

Inheritance is the process of creating a new class from an existing class. The new class inherits all the data and code from the existing class and can add its own data and code. This is a powerful way to reuse code and create new classes related to existing classes.

Polymorphism

Polymorphism is the ability of an object to take on different forms. This means the same code can work with different objects, even if those objects have different data or codes. Polymorphism is a powerful way to write flexible and easy-to-maintain code.

A brief history of object-oriented programming

The history of object-oriented programming is closely tied to the history of object-oriented languages. The first object-oriented language, Simula, was designed by Ole-Johan Dahl and Kristen Nygaard and released in 1967. Simula is considered the first truly object-oriented programming language, as it was the first language to allow objects to be created that could contain data and code.

Smalltalk, created by Alan Kay, Dan Ingalls, and Adele Goldberg in the 1970s, was a direct descendant of Simula. Smalltalk is an important milestone in the history of OOP, as it was the first language to be completely object-oriented, meaning that everything in the language was an object, including numbers, strings, and booleans. Smalltalk was also the first language to introduce the concept of inheritance, which allows new objects to be created by extending existing objects.

C++, created by Bjarne Stroustrup in the early 1980s, was the first language to combine the features of Simula and Smalltalk and add them to the C programming language. C++ is still used today and is widely considered the most successful object-oriented language ever created.

Java, created by James Gosling in the early 1990s, was designed to be a platform-independent, object-oriented language that could be used for creating large-scale applications. Java is used extensively in the development of web applications and is one of the most popular programming languages in use today.

Read also: CodeBlocks Can't Find Compiler? Here Is the Solution!

Why is OOP used in software development?

OOP is used in software development for several reasons:

1. To make code more modular: OOP allows developers to create software components that can be used in other programs. This makes code more reusable and easier to maintain.
2. To make code more efficient: OOP languages like Java are designed to run more efficiently on modern computer hardware.

3. To make code more reliable: OOP languages like C++ are designed to reduce the number of programming errors.

4. To make code easier to understand: OOP languages like Smalltalk are designed to be easier to read and understand than traditional procedural languages like COBOL.

5. To make code easier to develop: OOP languages like Visual Basic make it easy to develop software with a graphical user interface.

6. To make code more portable: OOP languages like Java can run on any platform that supports the Java Virtual Machine.

7. To make code more scalable: OOP languages like C++ can be used to develop very large software systems.

What are the disadvantages of object-oriented programming?

Object-oriented programming has a few potential disadvantages that could make it less ideal for some projects.

- Object-oriented programming can be more complex than other types of programming. This can make it more difficult to understand and work with, especially for beginners.
- Object-oriented programming can be less efficient than other types of programming. This is because it can require more memory and processing power to keep track of all the objects and their interactions.
- Object-oriented programming can be less flexible than other types of programming. Changing or adding new features to an existing object-oriented program can be difficult.

Object-oriented programming has potential disadvantages that could make it less ideal for some projects. However, it also has many advantages that could make it the best choice for other projects. It is important to weigh the pros and cons of object-oriented programming before deciding whether or not to use it for a particular project.

Tags **OOP** **Trending CS Topics**

Share: Facebook Twitter Whatsapp

OLDER
AI vs Human Creativity

NEWER
How to Install CodeBlocks in Linux (Ubuntu)

Sakib Mahmud

Hello, there! I am Sakib Mahmud from Bangladesh. Currently, I am studying computer science and engineering. And I love to write articles.

YOU MAY LIKE

Show more

POST A COMMENT

0Comments

** Please Don't Spam Here. All the Comments are Reviewed by Admin.*

Post a Comment (0)