

Cheat Sheet

Introduction to OOP

Good Software

Before jumping into Object Oriented Programming, let's understand the word **Software**.

Software is an easily changeable tool/product that performs a specific task.

The ease of changing or making changes to the software is referred as its softness.



- A good software should keep the users happy, by delivering what they need.
- A good software should also keep the developers happy. Ease of making changes (softness) keeps developers happy. So it should be:
 - Easy to understand and make changes.
 - Easy to fix bugs and add new features within the scope.

Object-Oriented Programming System (OOPS) is a way of *approaching, designing, developing* software that is easy to change.

Note

Keep in mind that building software differs from solving the coding questions (that you are doing in practice).

A fundamental difference is that you move on to the next problem once you solve a coding problem. But with software, once you build it, you are often required to add new features and fix bugs that need you to make changes to the code written by you or your colleagues. So unlike the coding questions, with software, you have to keep working with the same code for a long time.

Therefore, ease of understanding (code-readability) and ease of making changes (code-maintainability) to the code become crucial in software development.

The techniques and concepts you learn in this topic are developed over time based on software developers' experiences to make working code easier.

OOPs

Object-Oriented Programming is a way of approaching, designing and developing software, so that the components of the software and the interactions between them resembles real-life objects and their interactions.

Proper usage of OOPS concepts helps us build well-organized systems that are easy to use and extend.

Describing Real-Life Objects

In Object Oriented Programming, we model software after real-life objects. To be good at modeling software after real-life objects, we should be able to properly describe them.

Let us try describing these real-life objects



Object 1



Object 2



The following description is a **bad way of describing**, as the information of an object scattered and unorganized.

Object 1 is a car and it has four tyres.
Object 2 is a dog and it has four legs.
Object 1 has four doors.
Object 1 can make sound.
Object 2, barks.
Object 1 is in blue color.
Object 2 is in brown color.

Organized Description

- In the below description we are grouping the information related to an object.

Object 1 is a car and it has four tyres.
Object 1 has four doors.
Object 1 can make sound.
Object 1 is in blue color.

Object 1 is in blue color.

Object 2 is a dog and it has four legs.

Object 2, barks.

Object 2 is in brown color

- In the below approach, we further organize the information into
 - What the object is?
 - What the object has?
 - What the object can do?

Object 1 is a car

Object 1 has

Four tyres

Four seats

Four doors

and so on ...

Object 1 can

Sound horn

Move

Expand 

The above description shows a simple framework where we describe object by specifying the properties that the object has and actions that the object can do.

Organized Description should be

- A clear separation of objects.
- A clear grouping of what object has and what it does.

[Submit Feedback](#)