

What is oop

(OOP) is a computer programming model that organizes software design around data, objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior[techtarget].

OOP focuses on the objects that developers want to manipulate rather than the logic required to manipulate them. This approach to programming is well suited for software that is large, complex and actively updated or maintained. This includes programs for manufacturing and design, as well as mobile applications. For example, OOP can be used for manufacturing system simulation software[techtarget].

Advantages of oop[geeksforgeeks]:

- 1- We can build the programs from standard working modules that communicate with one another, rather than having to start writing the code from scratch which leads to saving of development time and higher productivity.
- 2- OOP language allows to break the program into the bit-sized problems that can be solved easily.
- 3- The new technology promises greater programmer productivity, better quality of software and lesser maintenance cost.
- 4- OOP systems can be easily upgraded from small to large systems.
- 5- It is possible that multiple instances of objects co-exist without any interference, It is very easy to partition the work in a project based on objects, It is possible to map the objects in problem domain to those in the program.
- 6- The principle of data hiding helps the programmer to build secure programs which cannot be invaded by the code in other parts of the program.

- 7- By using inheritance, we can eliminate redundant code and extend the use of existing classes.
- 8- Message passing techniques is used for communication between objects which makes the interface descriptions with external systems much simpler.
- 9- The data-centered design approach enables us to capture more details of model in an implementable form.

Disadvantages of OOP[geeksforgeeks]:

- 1- The length of the programmes developed using OOP language is much larger than the procedural approach. Since the programme becomes larger in size, it requires more time to be executed that leads to slower execution of the programme.
- 2- We can not apply OOP everywhere as it is not a universal language. It is applied only when it is required. It is not suitable for all types of problems.
- 3- Programmers need to have brilliant designing skill and programming skill along with proper planning because using OOP is little bit tricky.
- 4- OOPs take time to get used to it. The thought process involved in object-oriented programming may not be natural for some people.
- 5- Everything is treated as object in OOP so before applying it we need to have excellent thinking in terms of objects.

Dart oop Example[scaler]:

// Abstraction: Shape is an abstract class.

```
abstract class Shape {
```

// Encapsulation: area and perimeter are accessible only through these methods.

```
double get area;
```

```
double get perimeter;
```

```
}
```

```
// Inheritance: Rectangle inherits from Shape.
```

```
class Rectangle extends Shape {
```

```
    final double width;
```

```
    final double height;
```

```
    Rectangle(this.width, this.height);
```

```
// Polymorphism: area and perimeter are implemented for Rectangle.
```

```
    @override
```

```
    double get area => width * height;
```

```
    @override
```

```
    double get perimeter => 2 * (width + height);
```

```
}
```

```
void main() {
```

```
    Shape shape = Rectangle(5, 10);
```

```
    print('Area: ${shape.area}');
```

```
    print('Perimeter: ${shape.perimeter}');
```

```
}
```