

Object Oriented Programming

There are many definitions in different sources. Some of them are like: Object-oriented programming is a method of programming based on hierarchy of classes, and well-defined and cooperating objects (Oracle, 2010a)

Object-Oriented Programming (OOP) is different from procedural programming languages in several ways. Everything in OOP is grouped as objects. OOP, defined in the purest sense, is implemented by sending messages to objects (IBM, 1995a)

Object-Oriented Programming (OOP) is a programming language model organized around “objects” rather than “actions” and data rather than logic (TechTarget, 2008)

So in other words, OOPs is a type of programming in which the programmers have to define not only the data type of a data structure but also the functions or operations to be applied to the data structure. By doing this, the data structure converts to an object including both data and functions. Java, Python, Visual Basic .NET, Ruby, Smalltalk and C++ are among the popular OOP languages. The OOP starts with the first step as identification of all the objects to be manipulated and how they identify each other. This is often called data modelling. Once the object has been identified, we have to label it as the object's class and also have to define the type of data contained in it and any logical sequences that can manipulate it. A distinct logic sequence is known as method. An instance of class is called an object. The object runs on a machine (computer). Computer instructions are provided by methods and the data is provided by the object characteristics.

OOPs Principles

OOP deals with objects, and to define it, we should know the types of objects which are described in class. Objects are instances of class. A class contains methods and attributes.

Attributes – are used to represent an object by using a dot, like: `anObject.someAttribute`; the attribute can be any type even the same as an object type.

Methods – are used to represent the working of an object like:
`anObject.someMethod(parameters);`

The difference between attribute and method is that an attribute can only contain a value or a state but a method can also do some functioning.

In the programming world, there is not any definition of OOP which can be accepted by all at same time but certain principles have been formed to help in understanding OOPs (Filimon, 2007a):

Modularity – it means to break the program code into different parts which can then easily be interlinked to make a well organized program.

Encapsulation – it is linked to modularity. According to this, everything except the interface of an object should be hidden and precisely located, so the actual implementation can easily be changed, with an improvement to the program security (Filimon 2007a).

Inheritance – this is the main reason in the fine working of OOP. It allows a class to have the same behaviour as another class and extend or remove that behaviour to provide special action for specific needs (IBM, 1995a)

Benefits of OOP

With the help of inheritance, a programmer can eliminate the redundant code and extend the use of available code.

Classes

Programs can be built from the working modules which interact with one another, rather than starting the code again from scratch. Due to this, lots of development time is saved and production is higher.

Because of data hiding, the programs are designed not to be touched by the codes in other parts of the program and this provides greater system security and avoids unintended corruption.

Multiple instances of an object can coexist without any interference.

Easy to break the work in parts based on objects.

Easily small to large systems upgradation of OO systems.

With the help of the concept of data classes, a programmer can create any new data type which is not already defined in the language.

Code reuse; polymorphism.

Errors are caught when compiling rather than at runtime.

Works like the way the real world works. Easily to map a real world problem to a solution in OO code.

Disadvantages of OOP

OOP takes more time to execute as it is a high level concept and many routines run at the back at the time of execution.

Artificial class relations. We can easily make or break the program making it more suitable according to us but if the programming is based on real world problems then the modified structure is not similar to what exists in real life.

Unnecessary complications.

Wastage of time in case of small coding programs.

Due to inheritance, it is hard to read the code. If u have classes inheriting other classes, then you might end up with code again and again to figure out which method is called.

How OOP is used in Software Development?

In any complexity of software development, OOP is the best to solve the issue. These are the areas where OOP is used (Buyya et al, 2009):

Image processing

Pattern recognition

Computer assisted concurrent engineering

Computer aided design and manufacturing

Computer aided teaching

Intelligent systems

Distributed computing and applications

Component based applications

Business process reengineering

Enterprise resource planning

Data security and management

Mobile computing

Data warehousing and data mining

Parallel computing

OOP just helps us to load the real world problem into software which can be modified in different ways depending on the needs. Once a code generated for any software can be used again and again to serve other applications with the same functions. This helps in saving lots of time for the programmers and also helps in making the program flexible so that it can be easily modified when there is need.

I think OOP is not a successful approach to a beginner in programming because programming is not easy until the programmer does not know what objects or classes are. OOP has many advantages but they will work for the programmers having good knowledge in the concepts of programming. Despite all the advantages, a beginner cannot distinguish between the classes, objects or any other things which makes it difficult to understand. So I think a beginner should start with low level programs to learn the basic concepts and then move to OOP.

References

Buyya, R., Somasundaram, SS. and Chu, X. (2009) Object Oriented Programming with Java: Essentials and Applications. New Delhi, India: McGraw-Hill.

Filimon, T. (2007a) Object Oriented Programming [online]. Available from: <http://teodorfilimon.com/articles/OOP/Object-Oriented-Programming.htm>. [Accessed 14th January 2011].

Filimon, T. (2007b) Object Oriented Programming [online image]. Available from: <http://teodorfilimon.com/articles/OOP/Object-Oriented-Programming.htm>. [Accessed 14th January 2011].

IBM. (1995a) Introduction: What is Object-Oriented Programming? [online]. Available from: <http://www.inf.ufsc.br/poo/smalltalk/ibm/tutorial/oop.html#oop>. [Accessed 12th January 2011].

IBM. (1995b) Introduction: What is Object-Oriented Programming? [online image]. Available from: <http://www.inf.ufsc.br/poo/smalltalk/ibm/tutorial/oop.html#oop>. [Accessed 14th January 2011].

Oracle. (2010a) essentials of the Java Programming Language: A Hands-On Guide part 2 [online]. Available from: <http://java.sun.com/developer/onlineTraining/Programming/BasicJava2/oo.html>. [Accessed 12th January 2011].

Oracle. (2010b) essentials of the Java Programming Language: A Hands-On Guide part 2 [online]. Available from: <http://java.sun.com/developer/onlineTraining/Programming/BasicJava2/oo.html>. [Accessed 14th January 2011].

TechTarget. (2008) Object Oriented Programming [online]. Available from: <http://searchsoa.techtarget.com/definition/object-oriented-programming>. [Accessed 12th January 2011].