

Procedural programming languages and object oriented programming language



Procedural programming languages and object oriented programming language Introduction: Object oriented languages and procedural programming languages are basically two different paradigms based on two different thought processes. Both paradigms offer different approaches to model system solutions. Procedural programming is simply an approach to programming and the languages that support this methodology are called procedural programming languages, as is the case with object oriented programming languages. However it should be noted that it is possible to write an object oriented code using a procedural language like turbo C, similarly it is also possible to write a procedural code using an object oriented programming language like C#.

2. Procedural programming languages:

Procedural programming languages are those that follow the procedural paradigm for programming as is evident. The procedural approach basically guides to transform processes into procedure. First the processes are identified, and then they are broken down into smaller processes which finally shape into procedures. There are several limitations in the procedural approach like extension and re-usability of code is quite limited because of its nature as compared to the object oriented approach, where objects can be reused, updated and redesigned without having adverse effects on overall system functionality. Furthermore, code management also becomes challenging because of its sequential nature.

3. Object Oriented Languages:

Several object oriented languages offer program solutions based on the object oriented paradigm like Java, C++, . Net, Ada, SmallTalk etc. One of the most important aspects of OOP languages is dynamic binding or late <https://assignbuster.com/procedural-programming-languages-and-object-oriented-programming-language/>

binding.

OOP offers a rather different approach than procedural approach. The process of abstraction is managed by objects. Every process will be broken into inter object interactions. OOP languages supports objects, classes, instances, method, message, inheritance, subtype principle etc (Aaby Pg. 968). Class inheritance allows the member attributes and functions of one class to be used as if they were members of a second class. Inheritance is one of the primary characteristics of OO programming (Lippman, Pg. 391). In OOP methodology a system basically consists on interactions of objects and each object has to perform its own responsibility after that the responsibility is delegated to other objects. This delegation of responsibility is through the messages. In OOP methodology the system analysis and design consists of the following steps:

a. Object Identification:

Identification of important objects from the given scenario is the first step in abstraction. Which objects should be used within the system and which should not is a key point.

b. Object type definition i. e. classes:

Object definition or classes basically describe the objects. The class or object definition consist of properties/ attributes and behaviors/functions of the objects. Which attributes and behaviors are important is always challenging to identify.

c. Inter object relations:

Inter object relations are using and has relationship. Some times one object is a kind of other objects and some times one object uses the other object.

4. Discussion and conclusion:

<https://assignbuster.com/procedural-programming-languages-and-object-oriented-programming-language/>

Both the programming paradigms have their own pros and cons, procedural programming languages follow step by step development of a code which combine together to function as a program in contrast OOP approach provides a higher level of abstraction. One important aspect of OOP methodology is its reusability as compare to procedural languages (Lewis, 91). OOP techniques achieved reusability through encapsulation of program and data (Kim, Pg. 5). Furthermore, distribution of procedural codes over network based applications is a challenge and is now considered obsolete by most people. Most of the network/distributed applications use OOP methodology in which, only objects are supposed to communicate with each other without going into the details of either object. Java RMI, CORBA standards, . NET remoting all based on OOP methodology. However, remote procedure calls or RPC are being used in a very limited capacity.

5. Reference:

Aaby Anthony. (1996) Introduction to Programming Language. eBook

Lewis, J. et. (1991) An empirical study of the object-oriented paradigm and software reuse

ACM SIGPLAN, OOPSLA91 Conference, vol 26 issue 11

Kim W., Lochvosky F. (1989)Object-oriented Concepts, Databases and Applications. ACM Press Frontier Series

Lippman S. (1991) C++ Primer. 2nd Edition Addison Wesley Publisher.