

Q2.(a) What is object-oriented programming? How is it different from procedure-oriented programming? MDU BCA 2016, 2014

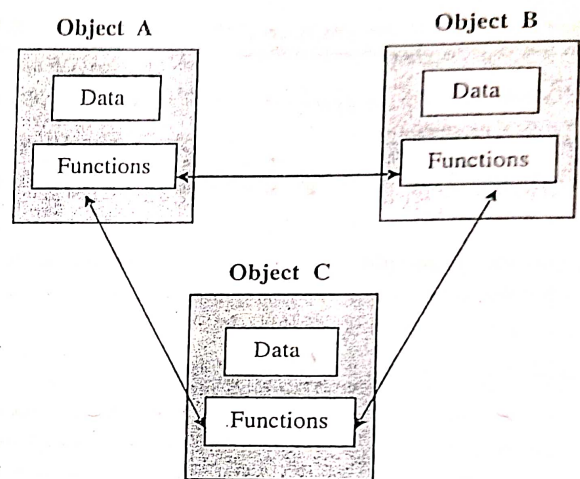
Ans. Object-Oriented Programming

Object oriented programming (OOP) is an approach that:

- Binds the data and functions that operate on the data into a single entity. Such an entity is called an object.
- Protects data from accidental modification from outside functions.

The object oriented programs are often easier to understand, correct and modify. This style of programming helps to bring the modularity in a program and ultimately the program becomes more productive. Any application can be defined in terms of entities or objects and hence it is called object oriented programming.)

OOP allows decomposing a problem into a number of objects and then builds data and functions around these objects. Functions of an object can only access its data. One can't access the data directly. Object-orientation is a concept which makes developing of projects easier. Thus object oriented programming attempts to solve the problems with only one approach – dividing the problems in sub-modules and using different objects. Objects of the program interact by sending messages to each other. The figure below illustrates this clearly:



(Object oriented programming, first developed in the 1960s) Programming with objects is quite like working with real-world objects. It groups operations and data into modular units called objects. These objects can be combined into structured network to form a complete program.

(Simula was the first object-oriented programming language. Java, C++, Visual Basic, .NET and Ruby are the most popular OOP languages today.)

The Java programming language is designed especially for use in distributed applications on corporate networks and the Internet. Ruby is used in many Web applications. Smalltalk, Delphi and Eiffel are also examples of object oriented programming languages.)

Object oriented programming is programming which is oriented around objects, thus taking advantage of Encapsulation, Polymorphism and Inheritance to increase code reuse and decrease code maintenance.

(For object oriented programming, an object oriented language is needed. A language can only be said to be object oriented if it supports encapsulation, inheritance and polymorphism.)

Basic Concepts of OOPS

Class: A class is a prototype that defines the variables and the methods common to all objects of a certain kind. Thus a class is a collection of objects of similar type. Once a class is defined, any number of objects can be created which belong to that class.

Object: Object is an instance of a class. Objects are the basic runtime entities in an object-oriented system.

Encapsulation: The act of placing data and the operations that perform on that data in the same class is called encapsulation. The class then becomes the 'capsule' or container for the data and operations. Data cannot be accessible to the outside world and only those functions which are stored in the class can access it.

Inheritance: Inheritance is the process by which objects can acquire the properties of objects of other class. In OOP, inheritance provides reusability, like, adding additional features to an existing class without modifying it. This is achieved by deriving a new class from the existing one. The new class will have combined features of both the classes.

Polymorphism: Polymorphism means the ability to take more than one form. This means that different classes may contain the same method names but the result which is returned by each method will be different as the code behind each method is different in each class.

Features of Object-Oriented Programming

Some of the main features of object-oriented programming are:

- *Object as a single unit combines both data and functions.*
- *Functions of an object can only access its data.*
- *Emphasis is on data rather than procedures.*
- *One can't access the data directly. It means, data is well secured from outside changes.*
- *It follows bottom-up approach in program design.*
- *Objects communicate with each other through functions.*
- *New functions and data can be easily added as per requirement.*

Difference between object-oriented programming and procedure-oriented programming:

Object-Oriented Programming	Procedure-Oriented Programming
1. It is data-oriented approach.	1. It is function-oriented approach.

Object-Oriented Programming	Procedure-Oriented Programming
2. Data and functions are combined together.	2. Data and functions are not combined together.
3. Data is secured.	3. Data is not secured.
4. It follows bottom up design approach.	4. It follows top down design approach.
5. An abstract data type can be defined which describes all objects of a class.	5. An abstract data type cannot be defined.
6. In this approach, objects exchange messages with each other.	6. In this approach, data is exchanged between procedures under the control of main program.
7. New data and functions can be easily added.	7. New data and functions can not be easily added.
8. The functionality of an existing class can be easily extended using inheritance.	8. It does not support the concept of inheritance.
9. It creates objects and communicates by sending messages to them.	9. It communicates by parameters and return values.

Object-Oriented Programming	Procedure-Oriented Programming
10. All features of procedure oriented programming (POP) may be present in object oriented programming (OOP).	10. It does not support the features of OOP like dynamic binding, polymorphism, inheritance etc.
11. A good programming approach for real time software design.	11. A good programming approach for scientific application development.

Q2.(b) Explain the characteristics of OOPs.

MDU BCA 2015

Ans. An object oriented language is one that uses object oriented programming techniques including inheritance, modularity, polymorphism and encapsulation.

Following are the important common features supported by object-oriented languages:

1. Encapsulation
2. Inheritance
3. Polymorphism
4. Abstraction
5. Message passing

1. Encapsulation

Data encapsulation, sometimes referred to as data hiding, is the mechanism whereby the implementation details of a class are kept hidden from the user.

The user can only perform a restricted set of operations on the hidden members of the class by executing special functions commonly called methods.

The concept of data encapsulation is supported in C++ through the use of the public, protected and private keywords which are placed in the declaration of the class.

Anything in the class placed after the public keyword is accessible to all the users of the class.

Elements placed after the protected keyword are accessible only to the methods of the class or classes derived from that class.

Elements placed after the private keyword are accessible only to the methods of the class.

Encapsulation enhances the data security and under this feature data cannot be easily corrupted as it is not easily accessible.

2. Inheritance

Inheritance is the mechanism whereby specific classes are made from more general ones. The child or derived class inherits all the features of its parent or base class and is free to add features of its own.

In addition, this derived class may be used as the base class of an even more specialized class. Inheritance provides a clean mechanism whereby common classes can share their common features rather than having to rewrite them.

3. Polymorphism

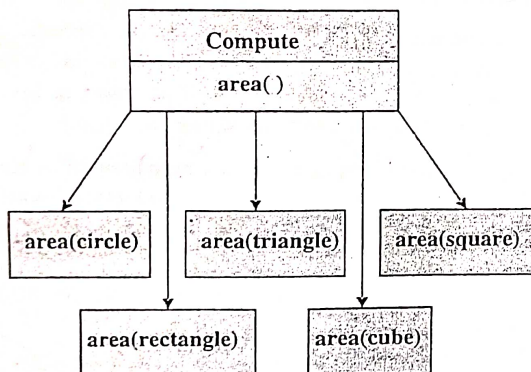
Polymorphism ("Poly" means "many" and "morph" means "form") means the ability to take more than one form. Polymorphism is the property by which the same message or data can be sent to objects of several different classes.

The operators or functions are used in different ways, depending on what they are operating on. It is the concept that supports the capability of an object of a

class to behave differently in response to a message or action. In the following example, same message is sent to objects of different classes and the objects behave differently.

In C++ one can declare many functions with a same name. For example, a function name `area()` can be used to compute area of a circle, rectangle, triangle, square, cube etc.

Polymorphism is often considered the most powerful feature of an OOP language.



4. Abstraction

Abstraction refers to the act of representing essential features without including the background details or explanations. For example, a person driving a car only knows how to drive a car i.e. the outer details but the

inner details about the working of different parts of machinery are hidden from the user.

5. Message passing

An object-oriented program consists of a set of objects that communicate with each other via messages. Objects interact and communicate with each other by sending messages to each other. When object A wants object B to perform one of its methods, object A sends a message to object B.

C++, Java, Smalltalk and C# are most popular object oriented programming languages. One can apply any of these given features in that language.