Q1. Define Object Oriented Programming Language?

Answer

Object-oriented programming (OOP) is a programming language model in which programs are organized around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior. Examples of an object can range from physical entities, such as a human being that is described by properties like name and address, down to small computer programs, such as widgets. This opposes the historical approach to programming where emphasis was placed on how the logic was written rather than how to define the data within the logic.

The first step in OOP is to identify all of the objects a programmer wants to manipulate and how they relate to each other, an exercise often known as data modeling. Once an object is known, it is generalized as a class of objects that defines the kind of data it contains and any logic sequences that can manipulate it. Each distinct logic sequence is known as a method and objects can communicate with well-defined interfaces called messages.

Simply put, 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 programs that are large, complex and actively updated or maintained. Due to the organization of an object-oriented program, this method is also conducive to collaborative development where projects can be divided into groups. Additional benefits of OOP include code reusability, scalability and efficiency.

Q2. List down the Benefits of OOP?

Answer

The Benefits of OOP offers to its users.

1. Re-usability

It means reusing some facilities rather than building it again and again. This is done with the use of a class. We can use it ‘n’ number of times as per our need.

2. Data Redundancy

This is a condition created at the place of data storage (you can say Databases)where the same piece of data is held in two separate places. So the data redundancy is one of the greatest advantages of OOP. If a user wants a similar functionality in multiple classes he/she can go ahead by writing common class definitions for the similar functionalities and inherit them.

3. Code Maintenance

This feature is more of a necessity for any programming languages, it helps users from doing re-work in many ways. It is always easy and time-saving to maintain and modify the existing codes with incorporating new changes into it.

4. Security

With the use of data hiding and abstraction mechanism, we are filtering out limited data to exposure which means we are maintaining security and providing necessary data to view.

Popular Course in this category

Sale

All in One Software Development Bundle (600+ Courses, 50+ projects)

600+ Online Courses | 3000+ Hours | Verifiable Certificates | Lifetime Access

4.6 (3,144 ratings)Course Price

$159 $1999

View Course

Related Courses

Python Certification Training (36 Courses, 12+ Projects)Java Training (38 Courses, 26 Projects)Programming Languages Training (41 Courses, 13+ Projects)

5. Design Benefits

If you are practicing on OOPs the design benefit a user will get is in terms of designing and fixing things easily and eliminating the risks (if any). Here the Object Oriented Programs forces the designers to have a longer and extensive design phase, which results in better designs and fewer flaws. After a time when the program has reached some critical limits, it is easier to program all the non-OOP’s one separately.

6. Better productivity

with the above-mentioned facts of using the application definitely enhances its users overall productivity. This leads to more work done, finish a better program, having more inbuilt features and easier to read, write and maintain. An OOP programmer cans stitch new software objects to make completely new programs. A good number of libraries with useful functions in abundance make it possible.

7. Easy troubleshooting

lets witness some common issues or problems any developers face in their work.

Is this the problem in the widget file?

Is the problem is in the WhaleFlumper?

Will I have to trudge through that ‘sewage.c’ file?

Commenting on all these issues related to code.

So, many a time it happens that something has gone wrong which later becomes so brainstorming for the developers to look where the error is. Relax! Working with OOP language you will know where to look for. This is the advantage of using encapsulation in OOP; all the objects are self-constrained. With this modality behavior, the IT teams get a lot of work benefits as they are now capable to work on multiple projects simultaneously with an advantage that there is no possibility of code duplicity.

8. Polymorphism Flexibility

Let’s see a scenario to better explain this behavior.

You behave in a different way if the place or surrounding gets change. A person will behave like a customer if he is in a market, the same person will behave like a student if he is in a school and as a son/daughter if put in a house. Here we can see that the same person showing different behavior every time the surroundings are changed. This means polymorphism is flexibility and helps developers in a number of ways.

It’s simplicity

Extensibility

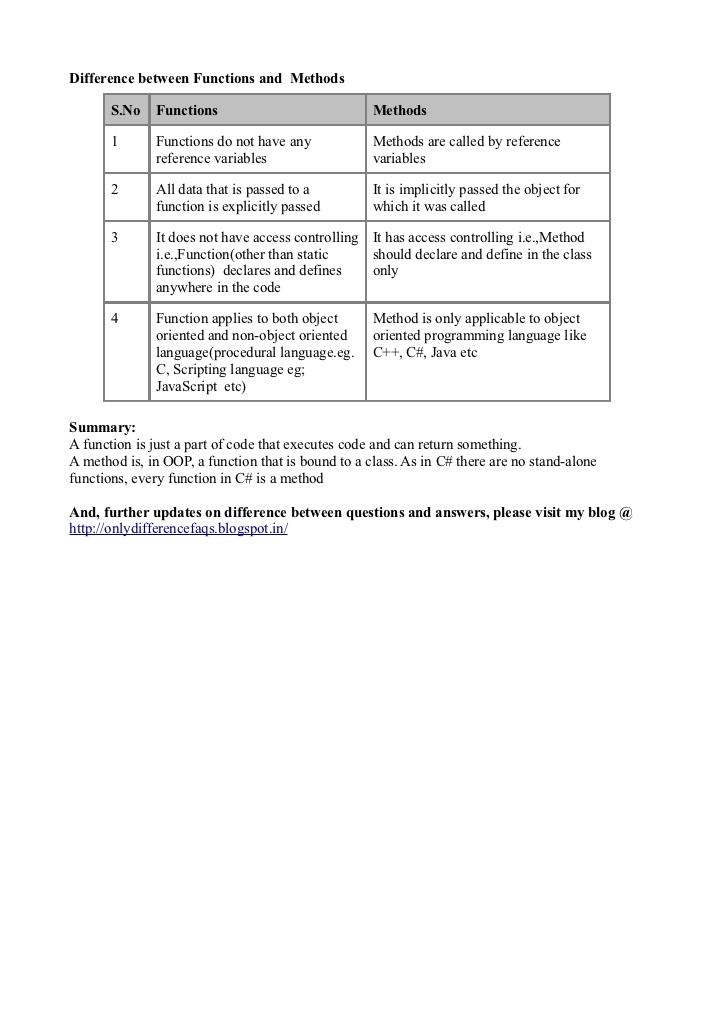
9. Problems solving

Decomposing a complex problem into smaller chunks or discrete components is a good practice. OOP is specialized in this behavior, as it breaks down your software code into bite-sized – one object at a time. In doing this the broken components can be reused in solutions to different other problems (both less and more complex) or either they can be replaced by the future modules which relate to the same interface with implementations details.

A general relatable real-time scenario – at a high level a car can be decomposed into wheels, engine, a chassis soon and each of those components can be further broken down into even smaller atomic components like screws and bolts. The engine’s design doesn’t need to know anything about the size of the tires in order to deliver a certain amount of power (as output) has little to do with each other.

Q3. Differentiate between function and method?

Anwer.



Q4. Define the following terms:

1. Class

2. Object

3. Attribute

4. Behavior

1. Class:

Python is an object oriented programming language.

Almost everything in Python is an object, with its properties and methods.

A Class is like an object constructor, or a "blueprint" for creating objects.

Create a Class

To create a class, use the keyword class:

Example

Create a class named MyClass, with a property named x:

class MyClass:

x = 5

1. Object:

Object is the blue print of class.

1. Atribute

Everything in Python is an object, and almost everything has attributes and methods. In python, functions too are objects. So they have attributes like other objects. All functions have a built-in attribute \_\_doc\_\_, which returns the doc string defined in the function source code. We can also assign new attributes to them, as well as retrieve the values of those attributes.

4.behavior

Objects in Python are generally classified according to their behaviors and the features that they implement. For example, all of the sequence types such as strings, lists, and tuples are grouped together merely because they all happen to support a common set of sequence operations such as s[n], len, etc. All basic interpreter operations are implemented through special object methods. The names of special methods are always preceded and followed by double underscores These methods are automatically triggered by the interpreter as a program executes. For example, the operation x + y is mapped to an internal method, add, and an indexing operation,, is mapped to getitem. The behavior of each data type depends entirely on the set of special methods that it implements.