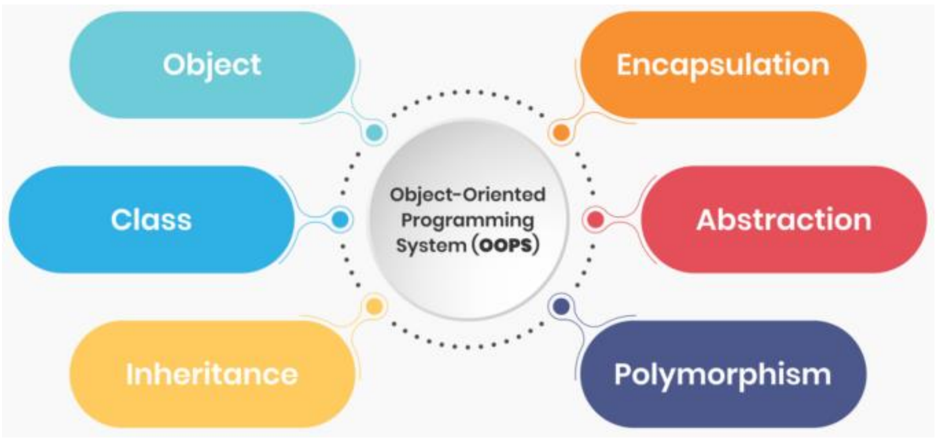
Object-Oriented Programming

1. Look at the diagram on the right. Are you familiar with the main notions of OOP? Can you define or explain each of them?



2. Match the terms related with OOP with their definitions.

1. **Object**	a) It is the mechanism of basing an object or class upon another object or class, retaining similar implementation.
2. **Class**	b) It is the concept of object-oriented programming that 'shows' only essential attributes and 'hides' unnecessary information.
3. **Method**	c) It is a specific realization of any object.
4. **Instance**	d) It binds data and its related methods together within a class. It also protects the data by making fields private and giving access to them only through their related methods.
5. **Template**	e) It is a template for a group of objects.
6. **Encapsulation**	f) It is the ability of an object to take on many forms.
7. **Abstraction**	g) It is a blueprint or formula for creating a generic class.
8. **Inheritance**	h) It is a unit of data that represents an abstract or a real-world entity.
9. **Polymorphism**	i) It can modify a class state that would apply across all the instances of the class.

3. Match the words in column A to their synonyms in column B.

A.	B.
attribute	influence
entity	spread
pillar	main principle
instance	external
diverse	case, example
extension	various
outer	characteristic
impact	object

4. Read the text about the main principles of OOP and get ready to answer the questions below.

a) What is the main idea of OOP?
b) How does OOP treat software development?
c) What is the difference between Object and Instance in OOP?
d) What can Method do? What activates Method?
e) What are the pillars of OOP?
f) What do Abstraction and Encapsulation have in common?
g) What does Inheritance help to avoid?
h) What does Polymorphism allow to do?
i) List the main benefits of OOP.

Main Principles of OOP

Object-oriented programming is one of the main programming methodologies, which is based on the idea that ***a program is a cluster of objects, each belonging to a certain class and the classes build up an inheritance hierarchy***. OOP model organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.

Object-Oriented-Programing allows programmers to think of software development as if they are working with real-life entities. In your everyday life, people have the knowledge and can-do various works/tasks. In OOP, objects have fields to store knowledge/state/data and can-do various methods.

Before diving into the four pillars of OOP, there are some basic terminologies to go over.

Object: The instance of a class. It's the working entity of a class. It's a unit of data that represents an abstract or a real-world entity.

Class: This is the model or standard about the capability of what an object can do. It is a template for a group of objects.

Method: It is a segment of code that defines an action. It can modify a class state that would apply across all the instances of the class.

Message: It activates Method.

Instance: It is like Object, however, within specific realization.

Now that we have covered these keywords, let's jump into the four principles of object-oriented-programming: **Encapsulation**, **Abstraction**, **Inheritance**, and **Polymorphism**.

The Four Principles of Object-Oriented-Programming (OOP):

Encapsulation

Diverse objects that are present under one program can automatically communicate with each other. If the developer wishes to stop this interaction, they will encapsulate every object individually in the form of classes. The process of encapsulation states that classes cannot communicate or change along with any particular variable and function of an object.

Abstraction

Abstraction is just like the extension of encapsulation as it can hide some properties and methods from being shared with outer code for making the interface of the object easy to understand. Developers specifically make use of Abstraction for different beneficial causes. In short, we can say that Abstraction helps in isolating the impact of changes made on the code for securing the inside code if something wrong happens and the particular change will only affect the present variables, not to the whole outer code.

Inheritance

Inheritance is the process of extending the existing code functionality for removing the repetitive coding work. The elements of HTML code contain a text box, a checkbox and a select field with some properties that are common with particular methods. In this, there is no need of redefining all the properties and methods for each HTML element as you can define all of them at once in a general object. Give a name to the object like 'HTMLElement' will help other objects to inherit all of its properties and methods and you can avoid the unnecessary work of repetitive coding.

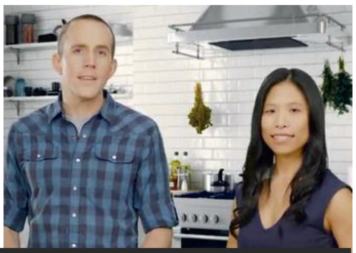
Polymorphism

Polymorphism means 'one name many forms' that allow developers to provide multiple elements depending on the object type. This will permit developers to redefine the whole work and define how it can be done by updating the parts in which it was previously performed. Polymorphism terms are known as overriding and overloading.

This approach to programming is well-suited for programs that are large, complex and actively updated or maintained. The organization of an object-oriented program also makes the method beneficial to collaborative development, where projects are divided into groups. Additional benefits of OOP include code reusability, scalability and efficiency. Even when using microservices, developers should continue to apply the principles of OOP.

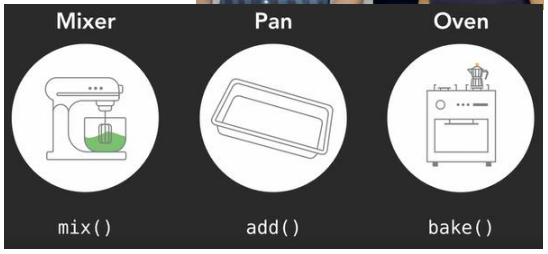
5. Address the text once again and identify the part the information below can belong to.

a) *Private:* Data that can only be accessed within the class.
Protected: Data that can only be accessed within the class, and its subclasses.
Public: Data or functions (methods) which can be accessed outside of the class.
b) It gives us a way to use a class exactly like its parent so there is no confusion with mixing types. This being said, each child sub-class keeps its own functions/methods as they are.
c) It is the process of selecting data from a larger pool to show only the relevant details to the object.
d) It is accomplished when each object maintains a private state, inside a class. Other objects cannot access this state directly, instead, they can only invoke a list of public functions. The object manages its own state via these functions and no other class can alter it unless explicitly allowed.
e) It is the ability of one object to acquire some/all properties of another object.



6. Watch a video episode devoted to understanding the difference between OOP and procedural code explained through cooking and indicate the following statements as True or False.

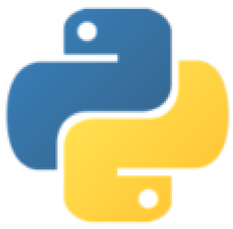
a) Object-oriented paradigm is very popular and widespread today.
b) OOP is always beneficial.
c) Procedural programming offers a straightforward approach.
d) Object-oriented code offers to create several mini programs where each object contains its own data and logic.
e) The end result of procedural approach and object-oriented one is not always the same.

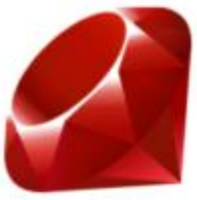


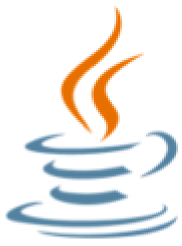
7. Watch the video again and get ready to match the two halves in the sentences below.

1. In procedural code, the program is written as	a) it's easy to think of simple programs in terms of sequential steps.
2. Programmers have a tendency to write code in this procedural manner because	b) similar to objects in the real world.
3. Instead of writing a single large program, my object-oriented code is	c) a long series of operations to execute.
4. We can talk about and use these programmed objects	d) code reusability.
5. One of the main advantages of using an object-oriented approach is	e) use them to write code in an object-oriented way or in a procedural way.
6. Object orientation is	f) split apart into several self-contained objects.
7. Multiple paradigms mean you can	g) referred to as a programming paradigm supported by many languages.

8. Each programming language has its own recognizable logo. Here are the logos of the most popular OO languages. Can you name them? Which in your opinion goes first?

a) 

b) 

c) 

9. Read about top three the most popular OO languages today. What makes them so attractive for programmers? Which one would you learn first and why?

1. **Java** is a general-purpose programming language that is class-based and object-oriented and designed especially for implementing all the possible dependencies. It allows developers to write once and run anywhere which means the compiled Java code has all the capabilities to run on every single platform that has the support of Java without needing recompilation.

Benefits of Java	***Drawbacks of Java***
An easy-to-learn language with clear syntax to understand. Java is platform-independent so you can write once and run anywhere. Java is a highly secure language. The multithreading feature allows us to write the program and perform multiple tasks simultaneously. A portable platform that has no implementation-dependent features. It is a high performing, interpreted language.	Slow and has poor performance because of the compilation and abstraction level of virtual machine. It offers an unattractive look and feel of the GUI. Java has no backup facility. It always requires a huge memory space. The codes of Java are verbose and complicated.

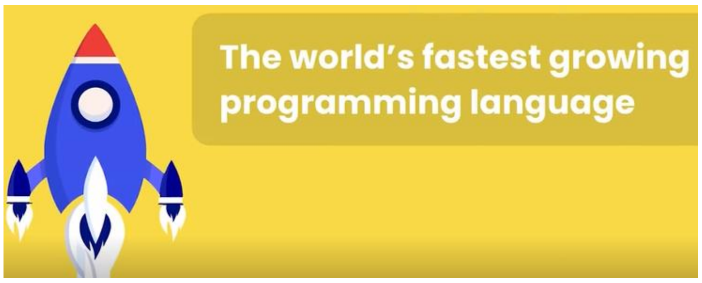
2. **Python** is an interpreted, general-purpose, high-level programming language. The design philosophy of Python focuses on code readability with the use of significant whitespace. Python’s language constructs as well as an object-oriented process for assisting developers in writing clear and logical coding for small as well as huge-scale projects.

Benefits of Python	***Drawbacks of Python***
It is easy to code and learn. Python is a free and open-source language that you can easily get from the official website. Python is the best option for crafting graphical apps. Python is a high-level language with extensible feature support. It is an integrated and portable language.	The speed of Python is slow as compared to C and C++. Not a good choice for mobile development. You should avoid selecting Python in memory-intensive tasks. It has limitations with database access. Python gives more run time errors and that’s why requires focusing on testing.

3. **Ruby** is an interpreted, general-purpose, high-level programming language. It can be dynamically typed and makes use of garbage collection. Also, it provides full support for programming paradigms that covers object-oriented, procedural, and functional programming. The language is specially focused on faster software development and that’s why called a ‘startup technology’.

Benefits of Ruby	***Drawbacks of Ruby***
Time-efficient language that reveals its potential in very little time. Ruby has many various helpful tools and libraries. It has a strong and active community. Ruby has a strong adherence to standards.	Ruby is less flexible in managing simple tasks. It provides no support for continuous updates and evolvement. The performance time of Ruby is very less especially when the project is big.

10. Watch a video episode about Python and choose the correct answer for each of the following questions. More than one option is possible in each case.



1. Who can use Python apart from software engineers?
   a) mathematicians
   b) scientists
   c) accountants
   d) networking engineers
   e) kids

2. What is Python used for?
   a) data analysis
   b) visualization
   c) automation
   d) building apps
   e) hacking

3. What makes Python so popular?
   a) It makes coding faster.
   b) It has simple syntax.
   c) It requires memory management.
   d) It is platform independent.
   e) Its community is vast.

11. Get ready to speak on the following topics.

- What are the main principles of OOP?
- What differs OOP from other paradigms?
- What tasks are suitable for OOP?
- Which OO languages are popular today?
- What are the main features of Python?
- How you can leverage using Python?