Difference between is-A and Has-A in OOP ?

IS-A Relationship:

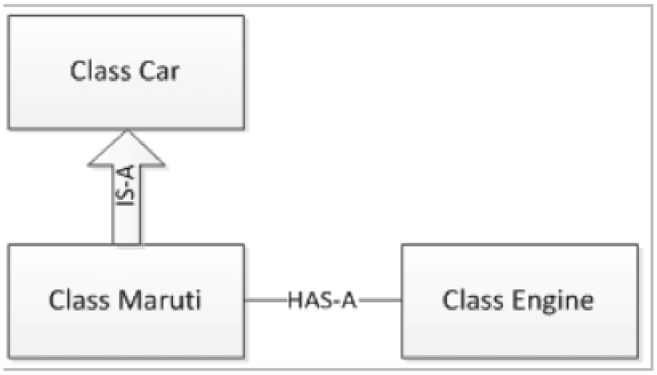
In object-oriented programming, the concept of IS-A is a totally based on Inheritance, which can be of two types of Class Inheritance or Interface Inheritance. It is just like saying "A is a B type of thing". For example, Apple is a Fruit, Car is a Vehicle etc. Inheritance is uni directional. For example, House is a Building. But Building is not a House.

It is a key point to note that you can easily identify the IS-A relationship. Wherever you see an extends keyword or implements keyword in a class declaration, then this class is said to have IS-A relationship.

HAS-A Relationship:

Composition(HAS-A) simply mean the use of instance variables that are references to other objects. For example, Maruti has Engine, or House has Bathroom.

Let’s understand these concepts with an example of Car class.



What is Super Harvard Architecture ?

The Super Harvard Architecture Single-Chip Computer (SHARC) is a high performance floating-point and fixed-point DSP from Analog Devices. SHARC is used in a variety of signal processing applications ranging from single-CPU guided artillery shells to 1000-CPU over-the-horizon radar processing computers. The original design dates to about January 1994.

Example of structure database and un structure ?

Examples of structural database are MMDB,PDB .

Examples of unstructured data are:

Rich media. Media and entertainment data, surveillance data, geo-spatial data, audio, weather data

Document collections. Invoices, records, emails, productivity applications

Internet of Things (IoT). Sensor data, ticker data

Analytics. Machine learning, artificial intelligence (AI)

what's dockerization, containerizations , Kubernetes, Jinkins ?

dockerization :

Dockizing is the process of packing, deploying, and running applications using Docker containers. Docker is an open-source tool that ships your application with all the necessary functionalities as one package. You can use Docker to pack your application with everything you need to run the application (such as libraries) and ship it as one package - a container. Containers are created from images that specify their precise contents. Dockizing is a big hit nowadays. All the big names use it - Google, VMware, or Amazon support it.

There are two parts of Docker:

The Docker Engine - a portable packaging tool

The Docker Hub - cloud service for sharing applications

The Docker container allows another user to quickly recreate the computer environment. The container provides operating virtualization by abstracting the “user space”. This technology ensures that your colleagues are developing or testing the product using the same environment as you which results in fewer errors.

Containerizations :

Containerization is a form of virtualization where applications run in isolated user spaces, called containers, while using the same shared operating system (OS). One of the benefits of containerization is that a container is essentially a fully packaged and portable computing environment.

Everything an application needs to run—its binaries, libraries, configuration files, and dependencies—is encapsulated and isolated in its container. The container itself is abstracted away from the host OS, with only limited access to underlying resources—much like a lightweight virtual machine (VM). As a result, the containerized application can be run on various types of infrastructure—on bare metal, within VMs, and in the cloud—without needing to refactor it for each environment.

Kubernetes :

This page is an overview of Kubernetes.

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.

Jinkins :

Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration purposes. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

With Jenkins, organizations can accelerate the software development process through automation. Jenkins integrates development life-cycle processes of all kinds, including build, document, test, package, stage, deploy, static analysis, and much more.

Jenkins achieves Continuous Integration with the help of plugins. Plugins allow the integration of Various DevOps stages. If you want to integrate a particular tool, you need to install the plugins for that tool. For example, Git, Maven 2 project, Amazon EC2, HTML publisher etc.

what's better pip or conda

pip

Python packages only.

Compiles everything from source. EDIT: pip now installs binary wheels, if they are available.

Blessed by the core Python community (i.e., Python 3.4+ includes code that automatically bootstraps pip).

conda

Python agnostic. The main focus of existing packages is for Python, and indeed Conda itself is written in Python, but you can also have Conda packages for C libraries, or R packages, or really anything.

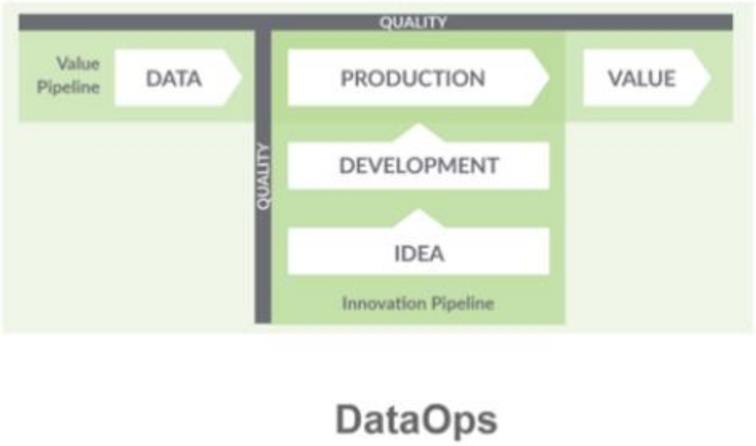
Installs binaries. There is a tool called conda build that builds packages from source, but conda install itself installs things from already built Conda packages.

External. conda is an environment and package manager. It is included in the Anaconda Python distribution provided by Continuum Analytics (now called Anaconda, Inc.).

conda is independently developed and supported outside of Anaconda. The conda-forge

Data ops cycle ?

DataOps is designed to build high quality data and analytics solutions at an increasingly accelerated pace with higher reliability.


QUALITY
Value Pipeline
DATA
PRODUCTION
VALUE
QUALITY
DEVELOPMENT
IDEA
Innovation Pipeline
DataOps