2) A process framework establishes the foundation for a complete software engineering process by identifying a small number of framework activities that are applicable to all software projects, regardless of their size or complexity. In addition, the process framework encompasses a set of umbrella activities that are applicable across the entire software process. A generic process framework for software engineering encompasses five activities-

Communication:
Before any technical work can commence, it is critically important to communicate and collaborate with the customer (and other stakeholders11 The intent is to understand stakeholders' objectives for the project and to gather requirements that help define software features and functions.

Planning:
Any complicated journey can be simplified if a map exists. A software project is a complicated journey, and the planning activity creates a "map" that helps guide the team as it makes the journey. The map —called a software project plan—defines the software engineering work by describing the technical tasks to be conducted, the risks that are likely, the resources that will be required, the work products to be produced, and a work schedule.

Modelling:
Whether you're a landscaper, a bridge builder, an aeronautical engineer, a carpenter, or an architect, you work with models every day. You create a "sketch" of the thing so that you'll understand the big picture—what it will look like architecturally, how the constituent parts fit together, and many other characteristics. If required, you refine the sketch into greater and greater detail in an effort to better understand the problem and how you're going to solve it. A software engineer does the same thing by creating models to better understand software requirements and the design that will achieve those requirements.

Construction:
This activity combines code generation (either manual or automated) and the testing that is required to uncover errors in the code.

Deployment:
The software (as a complete entity or as a partially completed increment) is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.

1) Simple:

- easy to learn, understand, and code
- Syntax is similar to C /C++.
- complicated features like pointers, operator overloading, structures, unions, etc. have been removed
- garbage collector it makes java more simple.

Secure:

- secure because it does not have pointers concept
- java provides a feature "applet" which can be embedded into a web application.
- Applet doesn't access to other parts of the computer, which keeps away from harmful programs like viruses and unauthorized access.

Portable:

- java programs to run on any computer or operating system.
- applet developed using java runs on a wide variety of CPUs, operating systems, and browsers connected to the Internet.

Object-oriented:

- In java, everything is an object.
- It supports all the features of the object-oriented programming paradigm.
- The primitive data types java also implemented as objects using wrapper classes
- it allows primitive data types to archive high-performance.

Robust:

- java has a strong memory management mechanism (garbage collector)
- java is a strictly typed language,
- it has a strong set of exception handling mechanism

Platform Independent:

- Java has invented to achieve "write once; run anywhere, any time, forever".
- JVM (Java Virtual Machine) achieve platform-independent.
- The JVM allows the java program created using one operating system can be executed on any other operating system.

Multi-threaded:

- Java supports multi-threading programming, which allows us to write programs that do multiple operations simultaneously.

Interpreted:

- Java enables the creation of cross-platform programs by compiling into an intermediate representation called Java bytecode.
- The byte code is interpreted to any machine code so that it runs on the native machine.

High performance:

- Java provides high performance with the help of features like JVM, interpretation, and its simplicity.

Distributed:

- Java programming language supports TCP/IP protocols which enable java to support the distributed environment of the Internet.
- Java also supports Remote Method Invocation (RMI), this feature enables a program to invoke methods across a network.

Dynamic:

- Java is said to be dynamic because the java byte code may be dynamically updated on a running system
- it has a dynamic memory allocation and deallocation (objects and garbage collector).