1) There are several JAVA Buzz words, Some of them are:

Simple:

- easy to learn, understand, and code
- some features like pointers, operator overloading, structures, etc. have been removed.

Secure:

- secure because it does not have a concept of pointers
- Applet doesn't access 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.

Object-oriented:

- In java, everything is an object, so, it supports all the features of the object-oriented programming paradigm.
- it allows primitive data types to get high-performance.

Robust:

- java has a strong memory management mechanism
- 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) achieves platform independence.

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 them into an intermediate representation called Java bytecode.

High performance:

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

Dynamic:

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

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. 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. 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 resources that will be required, and a work schedule.

Modeling: 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. 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 clear the 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.