1. Print Line

System.out.println() can print to the console
-> System is a class from the Core library provided by Java.
-> out is an object that controls the output
-> println() is a method associated with that object that recieves a single arguement.

```
System.out.println("Hello World!)
// o/p: Hello World!
```

2. Comments

Comments are bit of text, that are ignored by the compiler. They are used to increase the readability of the program.
-> Single line comments are created by :- //
-> Multi line comments are created by :- /* .... */

3. Compiling Java

In Java, when we compile the program, each individual class is converted into .class file, which is known as byte code.
The JVM (Java Virtual Machine) is used to run the byte code.

```
# compile the class file:
javac hello.java

# execute the compiled file:
java hello
```

4. Whitespace

Whitespace, including spaces and newlines, between statements is ignored.

5. Statements

In Java, a statement is a line of code that executes the task and is terminated with a semi colon (;)

```
System.out.println("Java Programming!!");
```

6. main() Method

In Java, every application must contain a main() method, which is the entry point for the application. All other methods are invoked from the main() method.

The signature of the method is public static void main(String[] args) { }. It accepts a single argument: an array of elements of type String.

```
public class Person {
  public static void main(String[] args) {
    System.out.println("Hello, world!");
  }
}
```

7. Classes

-> A class represents a single concept.
-> A Java program must have one class whose name is the same as the program filename.
-> In the example, the Person class must be declared in a program file named Person.java.

```
public class Person {
  public static void main(String[] args) {
    System.out.println("I am a person, not a computer.");
  }
}
```