**Operators , condtionals, loops , pattern programming**

**What is Turing complete language?**

Alan Turing created a machine that can take a program, run that program, and show some result. But then he had to create different machines for different programs. So he created "Universal Turing Machine" that can take ANY program and run it.

Programming languages are similar to those machines (although virtual). They take programs and run them. Now, a programing language is called "Turing complete", if it can run any program (irrespective of the language) that a Turing machine can run given enough time and memory.

For example: Let's say there is a program that takes 10 numbers and adds them. A Turing machine can easily run this program. But now imagine that for some reason your programming language can't perform the same addition. This would make it "Turing incomplete" (so to speak). On the other hand, if it can run any program that the universal Turing machine can run, then it's Turing complete.

Most modern programming languages (e.g. Java, JavaScript, Perl, etc.) are all Turing complete because they each implement all the features required to run programs like addition, multiplication, if-else condition, return statements, ways to store/retrieve/erase data and so on.

**switch – case**

break statement should be written in each switch case. Otherwise, once a case is matched, it executes all the remaining cases after it as well.