

1. Static RAM vs Dynamic RAM

S. No.	Static RAM	Dynamic RAM
1	It consists of transistors only.	It consists of transistors and capacitors.
2	SRAM are faster than DRAM.	DRAM are slower than SRAM
3	SRAM are used in cache memory.	It is used in main memory.
4	SRAM consumes less power than DRAM.	DRAM consumes more power than SRAM.
5	Its structure is complex.	Its structure is simpler than SRAM.
6	It consumes more physical space in computer system as compared to DRAM.	It consumes less physical space in computer system as compared to SRAM.
7	SRAM does not need periodic refreshment of data.	DRAM needs periodic refreshment to maintain the charge in the capacitor.

2. Compiler vs Interpreter

S. No.	Compiler	Interpreter
1	Compiler is a translator that converts the high level language (HLL) program into object code.	Interpreter is a translator that converts the HLL (unstructured programming/ structured programming) program into object code.
2	It checks the whole program at a time.	It checks the program line by line.
3	Compiler is not a good debugger.	Interpreter is a good debugger.
4	Its execution time is less as compared to interpreter.	Its execution time is more as compared to compiler.
5	Object code for the HLL program will not be created until all errors are modified.	It stops the execution of program whenever it found an error in a statement and display the error message on the screen promptly.
6	Programming language like C++ and C use compilers.	Programming language like Ruby, Python make use of interpreter.

3. Operator Associativity vs Operator Precedence

S. No.	Operator Associativity	Operator Precedence
1	When an expression has two operators with the same precedence, the expression is evaluated according to its associativity.	When two operators share an operand the operator with the higher precedence goes first.
2	Associativity can be either left-to-right or right-to-left, depends on the expression.	If two or more operators are involved in an expression, C language has a predefined rule of priority for operators.
3	Associativity means whether an expression like $x R y R z$ (where R is an operator such as + or <=) should be evaluated 'left-to-right' i.e. as $(x R y) R z$ or 'right-to-left' i.e. as $x R (y R z)$.	Precedence determines how an expression like $x R y S z$ should be evaluated (now R and S are different operators). If R has higher precedence than S, it will be evaluated as $(x R y) S z$, while if S has higher precedence than R it will be treated as $x R (y S z)$.