Data Types

- Data type defines storage space and format of variable.
- Primitive types
    - int - %d, %u, %o, %x
    - short - %hd, %hu
    - long - %ld, %lu
    - char - %c
    - float - %f, %e
    - double - %lf
- Integer types can be signed/unsigned
- Type qualifiers
    - const and volatile
- Derived types
    - Array
    - Pointer
    - Function
- User defined types
    - struct
    - union
    - enum
- void type – represent no value.
- Ranges of Datatypes
    - char
        - signed char (-128 to 127)
        - unsingned char (0 to 255)
    - int / long
        - signed int (-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647)
        - unsinged int (0 to 65,535 or 0 to 4,294,967,295)
    - short int
        - signed short (-32,768 to 32,767)
        - unsingned short(0 to 65,535)
    - long long
        - signed long (-9223372036854775808 to 9223372036854775807)
        - unsinged long(0 to 18446744073709551615)
    - float: ± 3.4E +/- 38
    - double: ± 1.7E +/- 308

Operators

- Types of operators
    - Arithmetic Operators (+ , - , * , / , %)
    - Assignment & shorthand Operators (= , += , -= , *= , /= , %= , &= , |= , ^= , ~= , <<= , >>=, ++, --)
    - Relational Operators (< , <= , > , >= , !=)

  - Logical Operators (&&, ||, !)
  - Conditional Operator (? :)
  - Bitwise Operators (& , | , ^ , ~ , << , >>)
  - Special Operator (, , sizeof(), [ ], * , & , □)

- Types of operators

  - Unary Operators (+ , - , ++, --, &, *)
  - Binary Operators (+, -, *, +=, ...)
  - Ternary Operators (? 😃

Arithmetic Operators

- Arithmetic operators work with all primitive data types i.e. int, float, char, double.
- Precedence of * & / is higher than + & -.
- % operator doesn't work with float and double type.
- % operator follows sign of numerator
- If two operands are of different types, the lower type is promoted temporarily for computation.
- char and short are promoted temporarily for computation.
- Char types are treated as integers (ASCII values) for calculation.
- If result exceed range of data type (overflow), then it rollback.

Relational and logical operators result in 0 or 1.

- 0 – indicate false condition
- 1 – indicate true condition
- Relational operators
  - <, >, <=, >=, ==, !=

Logical operators

```
* &&, ||, !
```

Comma operator

- evaluate to right-most value.
- have lowest precedence.

Control Statements

- Decision or Selection
  - if-else
  - switch-case
- Iteration (loop)
  - for
  - while
  - do-while

- Jump
    - break
    - continue
    - goto
    - return

if-else statement

- Condition is any expression – using relational, logical or other operators.
    - 0 – false condition
    - 1 – true condition

```
if (condition) {
    statement 1;
    statement 2;
}

if (condition) {
    statement 1;
    statement 2;
}
else {
    statement 3;
    statement 4;
}

// for block of single statement {} are optional
if (condition)
    statement 1;

if (condition)
    statement 1;
else
    statement 2;
```