-- MODULE 1 --

Day 1 --Installation of Java & Intellij IDEA

--IDE ? Integrated Development Environment

--Creating a new project in Intellij IDEA

--Exploring Settings of Intellij

--Font Size

--Theme

--Enabling zoom feature

--Writing your first program

--main public static void main(String[] args) { }

--sout System.out.println();

--Running a Java program

--Comments

--Single Line //

--Multi line /\* comments \*/

--Understanding the need of a variable

--Intro to data types

--int - stores integers Ex - 1 2 3

--String - stores combination of characters Ex - "Shery"

Day 2 --Variables

--can contain Data or Object References (DTL)

--Variable declration, Initialization

--Role of + operator between String & numbers

--String + String = String - Concatenation

--String + int = String - Concatenation

--int + int = int - Arithmetic Addition

--Naming Convention for Class/Variable/Method name - identifiers

--Must start with an alphabet or \_ or $

--Can end with a alphabet or \_ or $ or numeric digit

--Spaces are not allowed

--No reserved keyword

--Java is CASE SENSITIVE

--Cases and Conventions for clean and readable code.

--PascalCase - Class & Interface

--camelCase - variable and method name

--Game of brackets

--( ) - Methods - Parantheses

--{ } - Scope/body - Curly

--[ ] - Array - Square

--<> - Generics - Angular

Day 3 --Literal or constant

--Data Types - Graphic

--Primitive or Pre-defined

--Integer --why 4 ?

--byte, short, int, long( l or L suffix )

--Every Integer constant in java is int by default

--Floating Points

--Stop calling these decimal numbers

--float( suffix - f or F ), double( suffix - d or D - Optional)

--Every floating constant in java is double by default

--Non Numeric

--boolean, allowed values - true or false

--char - UNICODE (ASCII is a subset of UNICODE)

--Range - 0 to 65,535

--char can hold a equivalent int value

--Non Primitive

--User defined or Custom data types

--Derived from Primitive data types

--String literals

--String name = "Shery"

Day 4 --API - Application Programming Interface

--Reading Inputs from User

--Scanner Class

--Present inside java.util package

--Import Scanner class - import java.util.Scanner;

--Create Instance - Scanner sc = new Scanner(System.in);

--Standard Input, Output - STDIN, STDOUT

--Use methods to read respective data

--nextByte(), nextShort(), nextInt(), nextLong(), nextFloat(), nextDouble(), nextBoolean()

--Reading String Data

--nextLine() - Reads the whole line

--next() - Reads the first word

--Reading char data

--nextLine().charAt(0)

--Problem with nextLine method

--If nextLine is used right after any scanner method then it will not work properly.

--Remember to use a dummy nextLine method after reading nextXYZ data

which will pick the newline character

--Introduction to JAVA DOCUMENTATION

Day 5 --Operators in Java

--Unary, Binary

--Arithmatic +, -, \*, /, %, ++, --

--int/int will always yields int

--Modulo can work with int (works perfactly) as well as floats (produces unambiguity)

--Increment/Decrement operators can only be applied on variables, not on constants.

--Pre ++a

--Post a++

--Special powers of / & %

--/ to reduce the number

--% to get last digit(s) of number

--Relational <, >, <=, >=, ==, !=

--These returns values in boolean - true or false

--Logical &&, ||, !

--Used to combine multiple conditions

--ShortHand operators

--Precedence and Associativity of Operators

--Rest to be covered later like bitwise and shift operators

Day 6 --Control Flow Statements

--if(condition)

--executes its body if condition returns true

--works only on boolean values

--did not do anything if condition returns false

--if(condition) ... else

--executes body of if, if condition returns true otherwise executes else body

--else can't exist without if

--else do not have any condition

--if(condition) ... else if(condition)

--it is just a combitionation of the above two

--if(condition) if(condition) if(condition) if(condition) - Aka if ladder

--Nothing but a combination if multiple independent if statements

--Package

--Creating a new package

--package statement should be the first line in the java code file

--Used to group a similar set of classes (code management)

--Default Library package imported by default in every Java class

java.lang.\*

--Math class

--Present inside java.lang

--abs()

--floor()

--sqrt()

--cbrt()

--ceil()

--pow(double a, double b)

--round()

--max(double a, double b)

Day 7 --MORE PROGRAMS ON IF ELSE

--Discount on Bill

--INR Denomination

--Weekdays of corresponding number

--Ternary Operator

--<condition> ? true : false;

Day 8 --Loops

--Need of loops in programming

--Types

--Entry Controlled

--Exit Controlled

--for(init; condition; incre/decre)

--Executes its body untill condition returns true

--Syntax tweaks

--Init outside loop

--multiple conditions

--incre/decre inside loop body

--Conditions for infinite for loop

--for(;;)

--for(;;);

--if condition never returns false

Day 9 --while(conditions)

--Executes its body untill condition returns true

--Use it when the number of iterations are unknown

--Conditions for infinite while loop

--while(true)

--condition never returns false

Day 10 --Nested Loops

--Loop inside loop

--How to identify if nested loop is required ?

--Pattern Programming

--print() Vs println()

--Hackerrank

Day 11 --DO ... WHILE(condition);

--Executes at least one time

--Don't forget to put semicolon after while

--Executes untill conditions returns false

Day 12 --Switch statement

--Allowable data types - byte, short, char, int, String. (also wrapper classes of the same)

--Why not floating points ?

--Generates ambiguity (comparison is not perfact with floats)

--switch dont check each and every case.

--multiple cases can be combined together

--Seperate multiple cases by commas.

--cant use the expressions as cases.

--Generates ambiguity

--every case must be unique.

--fall through - gir jana.

--switch table

--Java v14

--Reducing boilerplate using arrow functions - aka lambdas

--No break statement required.

--Switch Expression

--yeild keyword

Day 13 --Arrays Data Structure

--Need

--Architecture

--One D array

--Different ways of initialization

--with new keyword

--With size

--int[] arr = new int[size]

--Without size

--int[] arr = new int[]{1, 2, 3}

--size and init can't be done together

--without new keyword

--int[] arr = {1, 2, 3}

--Default values of array element

integers - 0

floats - 0.0

char - null character, unicode - '\u0000'

boolean - false

non-primitives - null

Day 14 --Programs on array

Day 15 --Enhanced for

--Searching

--Linear Search

--Binary Search

--Sorting

--Bubble Sort

Day 16

--methods

--type conversion

--passing

--array pass

--leetcode intro

stack calls in a normal prog

heap

recursion

break

continue