```
Eclipse
=======
- Eclipse is a Java IDE
- IDE stands for Integrated Development Environment
- It is Open Source
- It provides many tools for developers

Shortcut Keys
=============
To generate the main method => main + (ctrl + space) + enter
To generate comments => Select the para + (Ctrl + /)
Ctrl + Space => Code suggestion
Ctrl + Shift + F => Code Formatting
Ctrl + Shift + T => Find a class
Alt + Shift + R => Refactor (Rename at all places)
See the location of project: Right Click on Project name -> Show In ->
System Explorer

Create project in Eclipse
=========================
Eclipse IDE configurations
--------------------------
1. Window -> Perspective -> Open Perspective -> Java

2. Tabs we need -> Project Explorer, Console, Problems
Window -> Show View -> Other -> [Select Tabs]

3. Increase the font-size -> Window -> Preferences -> General ->
Appearances -> Colors & Fonts

Project Hierarchy
-----------------
On the top, we have -> Project (P01CoreJavaPractice)
Inside Project, we have -> Packages (ex01.basics)
Inside package, we have -> Source code (Ex01FirstJavaProgram.java)

First Project
-------------
- Creating our 1st project: File -> New -> Java Project
- Project Name: P01CoreJavaPractice
- It should be in UpperCamelCase

First Package
-------------
- Right click on src -> New -> Package
- The package name should be in small caps
- Packages are nothing but folder structure
- ex01.basics will create a folder "ex01" and inside it folder named
"basics"

First Class
-----------
- Right click on package ex01.basics -> New -> class
- The class name should be in upper camel case (Ex01FirstJavaProgram)
```

```
- All the code will be inside this class apart from package, import &
comments

Note: See the location of project: Right Click on Project name -> Show In
-> System Explorer

Be thoughtful about:
1. Indentation
2. Eclipse Shortcut Keys
3. Saving the file after changes are made
4. Watch out for Warnings
5. Watch out for Errors
6. Reading Console
-----X-----X-----X-----
Operators
=========
6 + 7

Operand: The values on which operation is done is an operand. Here 6 and
7 are operands

Operator: The operation on operands is done by operator. Here + is the
operator

Types of Operators
------------------
Based on the number of operands, there are 3 types of operators:
1. Unary Operators
2. Binary Operators
3. Ternary Operator

Unary Operators
===============
There are 5 unary operators:
1. ++ (Increment operator)
2. -- (Decrement operator)

3. + (single additive operator)
4. - (single subtractive operator)
5. ! (not operator)

int i = 10;
++i; // pre-increment operator
i++; // post-increment operator
S.o.p(i);

int i = 10;
--i; // pre-decrement operator
i--; // post-decrement operator
S.o.p(i);

pre-increment operator (increment first, then assign)
post-increment operator (assign first, then increment)
```

```
int i = 10;
int j = i--;

S.o.p(i);
S.o.p(j);

Binary Operators
================
There are 4 types of Binary Operators
1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Assignment Operator

1. Arithmetic Operators: +, -, *, /, %

They work on numbers and return a number

2. Relational Operators: <, >, <=, >=, ==, !=

They work on numbers and return a boolean (true/false)

3. Logical Operators : &, |, !

They work on boolean and return a boolean

boolean pass = true;
boolean fail = !pass;

& - AND Operator => Both conditions must be true
----------------
True & True => True
True & False => False

False & True => False
False & False => False

| - OR Operator => Either conditions must be true
----------------
True | True => True
True | False => True

False | True => True
False | False => False

Short circuit logical operators : && and ||

A && B
------
If A is false, B will not be evaluated
If A is true, B will be evaluated

&& -> It will not evaluate the
expression on RHS if LHS is
```

```
false and the final answer will be false

A || B
----
If A is true, B will not be evaluated
If A is false, B will be evaluated

|| -> It will not evaluate the
expression on RHS if LHS is
true and the final answer will be true

Assignment Operator
===================
Simple Assignment Operator
--------------------------
SAO assigns the value of an expression on RHS into variable on LHS

int total = n1 + n2;

n1 + n2 = total; Wrong

Compound Assignment Operator
----------------------------
x = x op y;
x op= y

int num = num op 10;
num op= 10;

Ex:
int num = 20;

num = num + 10;

num += 10;

1. x = x + 10 => x += 10 => Compound Addition Operator
2. x = x - 10 => x -= 10 => Compound Subt. Operator
3. x = x * 10 => x *= 10 => Compound Mult. Operator
4. x = x / 10 => x /= 10 => Compound Quotient Operator
5. x = x % 10 => x %= 10 => Compound Modulus Operator

Ex:

salary = salary + bonus;
salary += bonus;

balance = balance - withdrawalAmount
balance -= withdrawalAmount

balance = balance + depositAmount
balance += depositAmount

Ternary Operator (?:)
```

```
====================
operand1 ? operand2 : operand3

Condition ? Expression 1 : Expression 2

Ternary operator is a short form of single if-else statement

int age = 19;
int eligibility = 0;

if(age >= 18) {
      eligibility = 1;
} else {
      eligibility = 0;
}

age = 20;
eligibility = age >= 18 ? 1 : 0;
-----X-----X-----X-----X-----X-----
```