Chapter 2 :-

Writting Program Steps : IPO

1. State your Problem what is
2. IF it's a huge Problem divide it into Small Problems
3. Design an algorithm to solve each Problem only
4. collect all the Solution to form the Solution of your main problem

Note: Every Programs in java must contain an Entry point called main Method

---

Reading input from the user :

we import from Package called java.util a class called Scanner by the following Syntax :

```
import java.util.Scanne ;
```

and we must make a declaration of an object from it by following Syntax :

```
Scanner objectname = new Scanner (System.in);
```

and we want to ask for value from Computer inputs System.in Method by the following Syntax :

nextDatatype(); Method

objectnam.nextDatatype(); for calling the Method

note we can use import java.util.*; To import all classes in java.util Package

---

Identifiers :

Def: The Identifier is the name of address ~~white~~ in which Data is Stored and when we want to name an Identifier we must follow the following Rules :

1. sequence of characters consists of letter, (_), $ can't start with digits
2. Identifier can't be a keyword from java Syntax
3. Identifier can't be True, false or null
4. Identifier Can be any Length

note If you have a wrong Identifier It is ~~Case~~ Cause Syntax E.

note all the following Identifiers are different area, Area, AREA and you can use that Syntax by large length numOfStds

Variables :

Def: is just as a box contains one value and have one specific letral and we declare a Variable by following Syntax :

Datatype variablename ;

and we have in java more than one datatype

* int : integer numbers : 32 bit → $\pm 2147483648 - 1^{+}$
* short : // : 16 bit → $\pm 32768 - 1^{+}$
* long : // : 64 bit → -2^{63} to $2^{63} - 1$
* float : floating point numbers : 32 bit
* double : // : 64 bit
* char : characters : 1 bit
* Byte : integer numbers 8 bit → -128 to 127
* boolean : True or false : Truth Value

and assign a Value for Variable by The following Syntax :

Passing ← assignment operator

Variablenam = i.Value ;

note: A Variable must be declared before it can be assigned a Value and when you be in method you must assign it it's initial value

ote: you can use these expressions s.o.p ln(x=1), i=k=j=1;

Named Constants:

Def: is a variable declared and initialized but you can't change it's value of initialization anymore and we use the following syntax:

final datatype CONSTANTNAME = i.value ; (i. → initial)

note all Constant Names are characters capitalized Id.

Naming Conventions:

1. use Lower case for Variables and methods
2. Capitalize first letter of classes names
3. Capitalize every letter of Constant Variables

Numeric Operators:

1. \+ → addition
2. \- → subtraction } Second
3. * → multiplication
4. / → Division } First
5. % → Remainder

note all Mathematical expressions are excuted from Left to Right

---

Literals :

Def: is an initialized form of any datatype Value

note To denote a binary integer Literal use a Leading of 0B or 0b
To denote an Octal integer Literal use a Leading of 0
To denote a hexadecimal integer Literal use a Leading of 0x or 0X

note Hexadecimal is 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
octadecimal is 0,1,2,3,4,5,6,7
Unicode is \uF01A
ASCII Code is 65, any number represents a character

and you can use some Scientific notations to round your number by following Syntax

number E ± 2 = number * 10^2

note when we want more readability use the following Syntax
long ssn = 232_45_4519; and the underscore must be placed between 2 digits

---

Operator Precedence :

( ) → *, ÷, %, → +, -

---

note

System.currentTimeMillis() Method that Returns The Current time from 1970 GMT in milli seconds

Augmented Operators :

1. `+=` → Addition Assignment `i = i+8; ≡ i+=8;`
2. `-=` → Subtraction Assignment `i = i-8; ≡ i-=8;`
3. `*=` → Multiplication Assignment `i = i*8; ≡ i*=8;`
4. `/=` → Division Assignment `i = i/8; ≡ i/=8;`
5. `%=` → Remainder Assignment `i = i%8; ≡ i%=8;`

Increament and Decreament :

1. Pre that is excuted then stored in the Variable
   `++a;` or `--a;`
2. Post that is stored then excuted then Stored the new Value
   `a++;` or `a--;`

Casting :

Def: is a way to change dataType of the Variable by following Syntax :

`(datatype) Variable name/it's Value;`

Common Errors and Pitfalls :

1. Undeclared , Uninitialized , Unused Variables Errors
2. Integer Overflow Error
   That the Value exceeds the Range or Permissible Range
3. Round off - Errors
   number of decimals out of the Premissible Range
4. Unintended Integer Division Error
   If two Operands are integers it will be Integer Division
   your Brain : $\frac{1}{2} = 0.5$
   Machine : $\frac{1}{2} = 0$ → neglicting decimals
5. Redundant Input Objects Error
   That you declared Two Objects to take two Inputs
