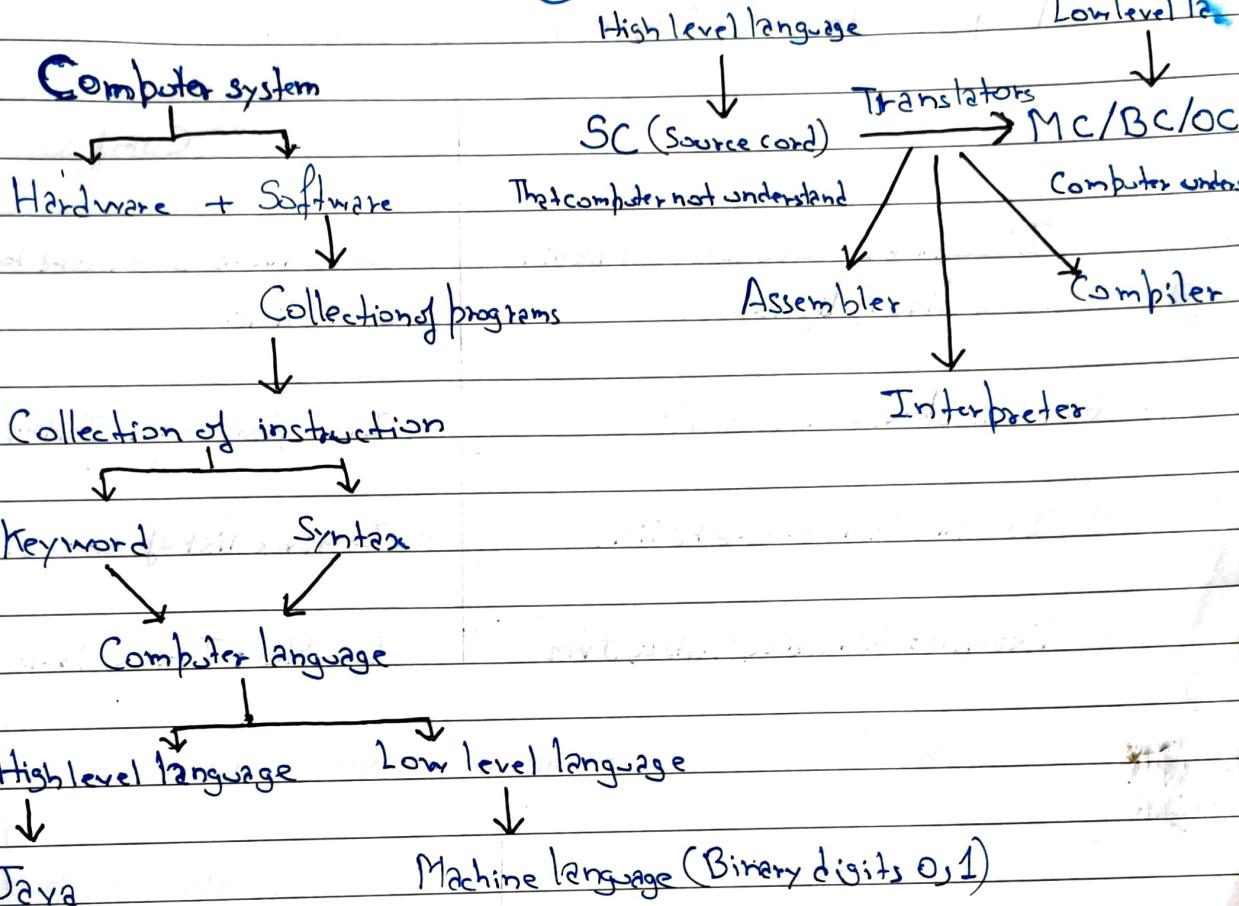


Java on BlueJ Environment



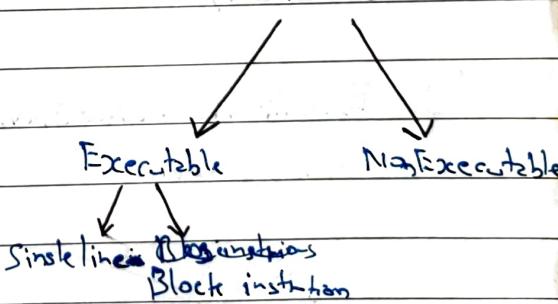
Computer only understand machine language

Benefits of Low level language:

- Directly understand computer
- Faster result

Problems of High level language:

- Machine language dependant
- Very Very Very difficult to rebug (rectify)
- Very Very Very difficult to write



For every symbol of HLL we need 8, 0 and 1 combination

$\therefore 8 \text{ bits} = 1 \text{ byte}$

1 byte = 1 character

Difference between Interpreter and Compiler:

Interpreter

1. It converts source code to machine code line by line.
2. It is slower than Compiler
3. It gives one error at a time
4. Examples: GWBASIC, QBASIC, JVM

Compiler

1. It converts source code to machine code at a time.
2. It is faster than Interpreter
3. It gives list of errors at a time
4. Examples: TurboC, TurboC++, Java C

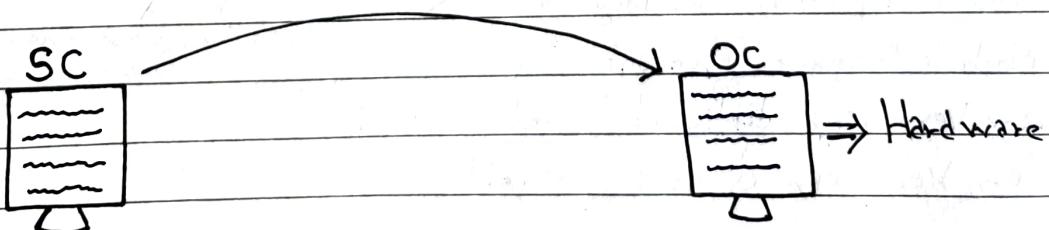
Difference between Source code and Machine code:

Source code

1. Code generated by the user
2. Understand by the user, but not by the computer
3. Extension is .BAS-JAVA as for the translator used.

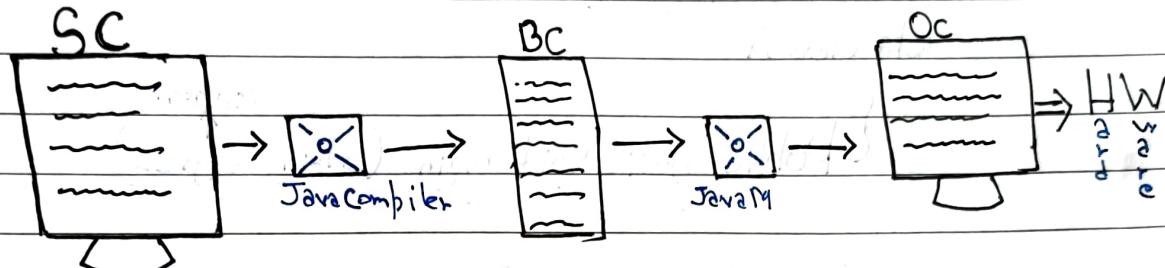
Machine code

1. Code generated by the computer (translator)
2. Understand by the computer but not by the user
3. Extension is .obj



C, C++, these software can directly be converted to object code. These software when creates a program can only run in that particular environment. E.g. when it is created in Linux environment, it will only

Work in Linux environment all over the world when transferred to other Linux software but not in other environment like Microsoft, Unix, etc.



JAVA can run on every environment as because the software is designed in such a manner that can run irrespective of architecture and platform of any computer. It is possible as byte code ask as a common code for all JVM machine.

WORA language: Write once run anywhere language.

Requirements for loading a JAVA program:

Two software are essential for loading of JAVA program environment in the computer are JDK (Java Development Kit) and BlueJ

Blue J: It is a software which act as an ^{write Save modify} (editor) and a debugger of JAVA language.
 ↴
 Convert to bytecode

How to write JAVA program.

/* To display sum of unit digit of 2 no's */

import java.lang.*;
class A

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

{ int a,b,ua,ub,s;

a=25

b=37

ua=a%10

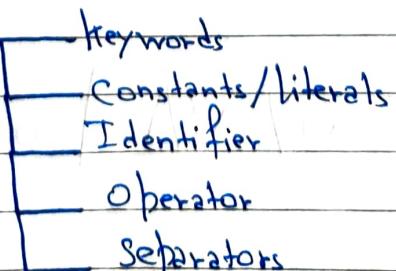
ub=b%10

s=ua+ub

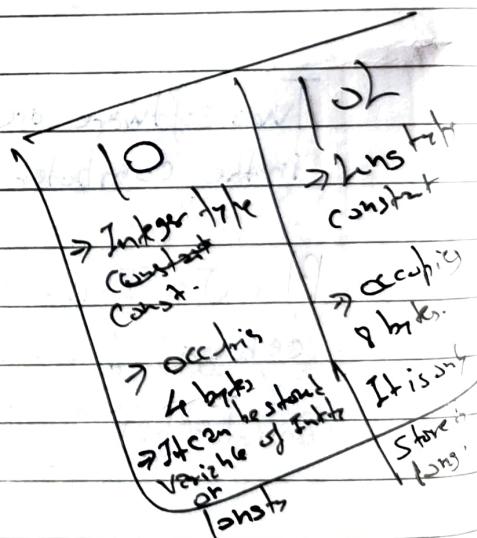
System.out.println ("The sum is "+s);

Conclusion

- (i) Use of class
- (ii) Use of Remark statement.
- (iii) Use of package
- (iv) Use of main() method
- (v) Use of Token



Curly bracket is used to create a block where execute instruction to be written.



3 ways to give remarks statements

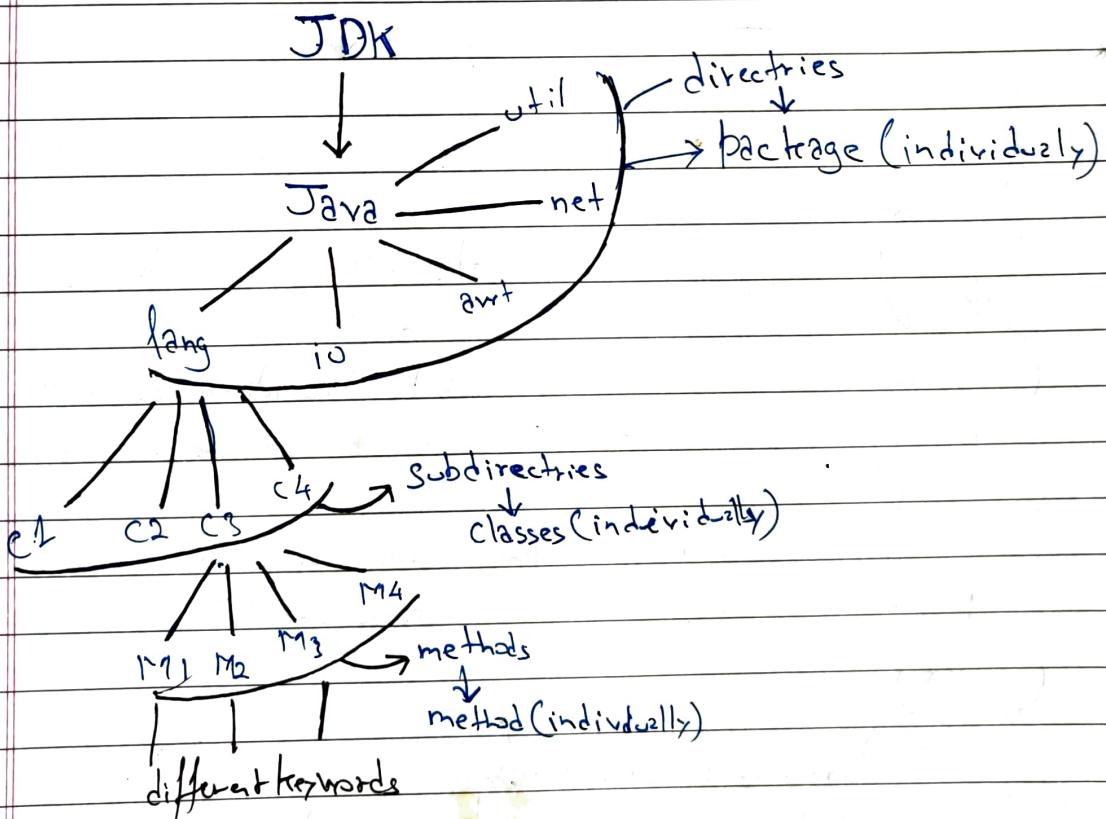
(1) // Single line (Executable only one line)

(2) /*

====
====
*/

Multiple line (Executable number of line)

(3) /** Documentation
===== Description variable
*/



Package is a collection of software.

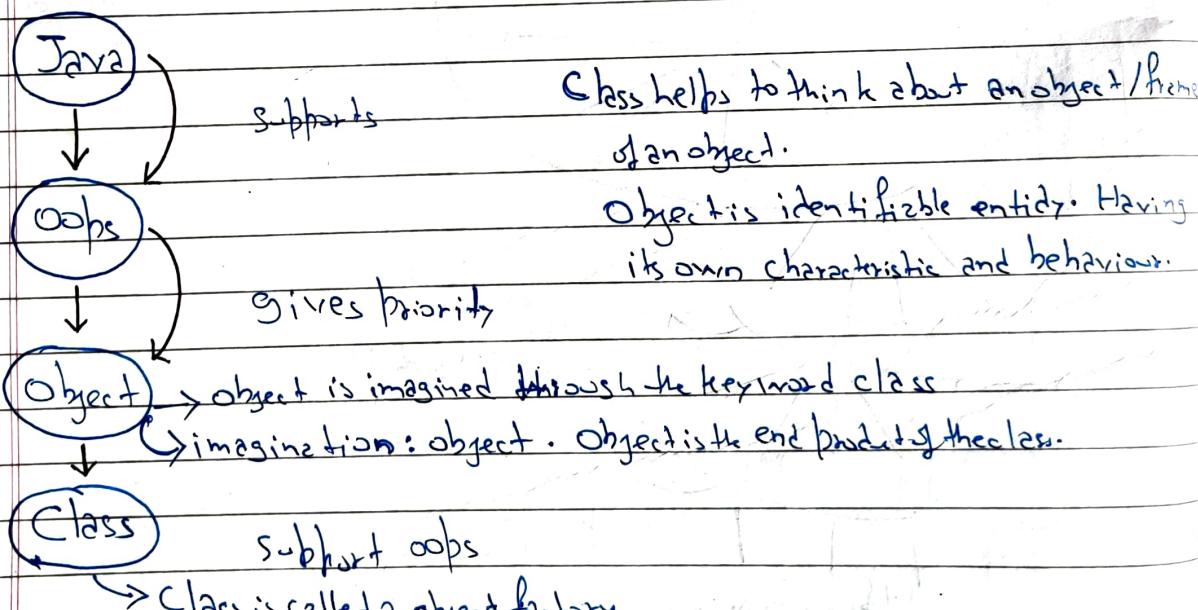
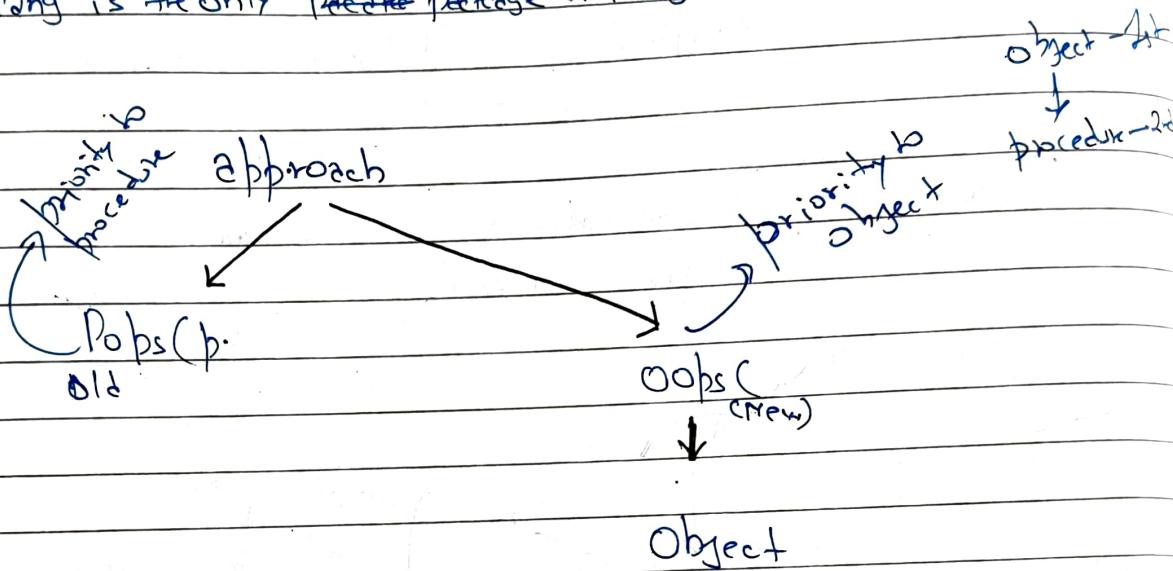
Packages is the collection of classes where keywords are defined

Import is a keyword help to activate the package (bring and keep it in H.R.A)
Activation means to bring the package from harddisk to RAM.

Block instruction: Start with curly bracket { and end with curly bracket }.

Single line: A single line which ends in semicolon.

long is the only ~~datatype~~ package which gets activated automatically.



Token: A smallest individual part of a program having its own identity.

When an imagination is created, class can create as many objects with the help of that particular imagination.

Attribute: state of an object.

"1|n") → 1n

Keywords

Constants / Literals

Identifier

Operators

Separators / Punctuators

"1" → 1"1" → >"1|DBSR1" → (1DBSR1) → 2 bits."B|'B|'S|R" → B'B'SR

Keyword: These are the reserved words used in JAVA program in order to write instructions. All the keywords are defined in different packages present in JAVA directory. It is important to note that all keywords are in lower case (small letters).

Examples of keywords:

in = keyboard, out = monitor

import, class, public, static, void, etc; in, out, etc else, true, etc.

There are 50 keyword used in JAVA.

All the keywords are small letter.

Constants: It is a part of a program which never change its worth or value during execution depending upon the nature there are five types of constants such as

1 ; "120" 12
 15 'A' → Alphanumeric

Numeric

→ Nonnumeric

Integer Decimal Real constant Character String Boolean

4-3

0.32

'A'

"ABC"

True

77-20

0.59

(F)

"1.25"

False

32,42

99,

All single
surve
are character
constant

'H'

"double"

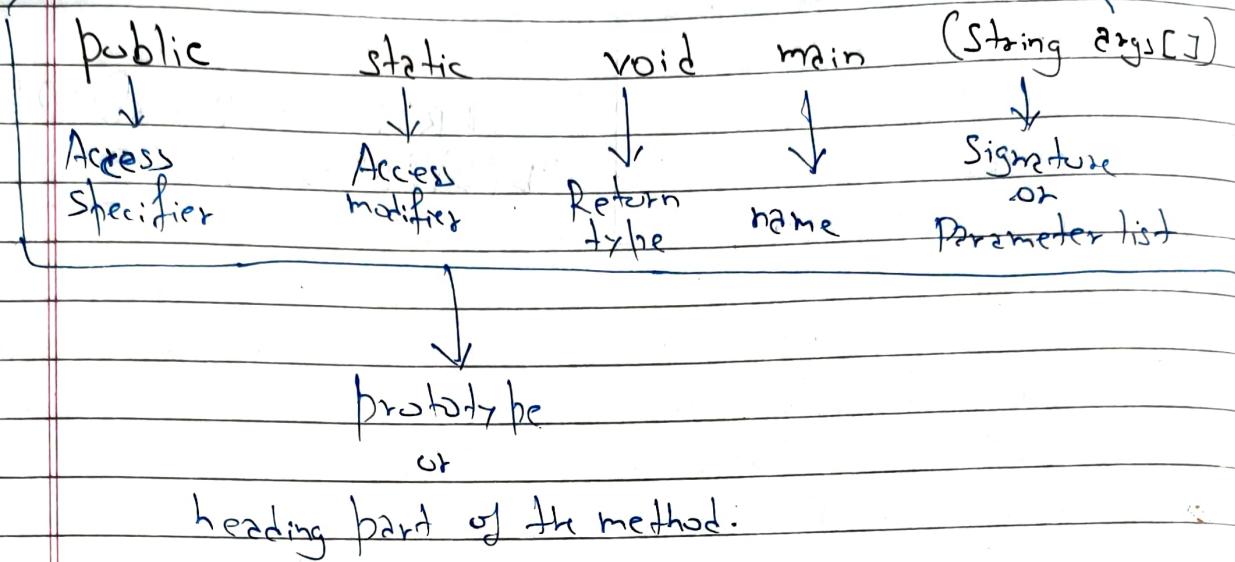
Java characters: 65536

→ Always occupy 16 bits

in single equal

More than one character is treated in " "

(Command line argument)



main: The main method from where compilation and execution starts. It controls the class. It is a most important method of a program, which must be present in the class.

public: The public method helps to use in other classes, available to other classes. Where it is available (Access specifier)

static: Static help to make method all object have access but no copies of the object. Static helps to create a common element which will be accessed by all the objects. (Access modifier) Only one piece

Void: The method type which cannot return the instruction taken from the OS.

String args[]: Input, took information from OS
(C:\>) → command line also called as command line argument

Identifier: It is a part of a program which help to recognize a block or variable.

Example of identifier: CN (Classname), MN (Method name), VN (Variable name) and LN (Local name)

It is the only part of a program which will given by the programmer.

In order to create identifier following rules should be followed called naming rules.

1. The valid symbols are: A to Z, a-z, 0-9, - (underscores), \$.

Example: A^L, A*B^X, A-B^V, A-BX, A BX

2. Digits cannot be in the first place.

Example 2A^X = A2^V

3. It is case sensitive. (Uppercase and lowercase are treated as different).

int A, a ✓ after capital letters

a=5 x because only accept capital letters.

Principle ✓

4. It can be of any length. The name should be small and easy to explain (explanatory). Must be familiar or known that help to remember.

Example J= P*R*T/100 (Easy to recognise)

a= b*c*d/100 (Hard to remember) → less accessible

5. Keyword cannot be identifier.

Naming convention: It is a concept used in JAVA in order

to distinguish the identifier

Rules for naming conventions.

1. If it is a class name first letter of each word Capital.

Example : Class name

For one letter A (Capital)

For a word Amount (First letter)

Princip... A... (Continue)

2. If it is method name or variable name first letter of subsequent word is Capital

Example

Method and Variable Name

a (Small)

Amount (Small value)

b.... Amount (Subsequent part are Capital)

3. If any variable is used to keep fixed value then its all alphabets will be Capital.

Ex: $\pi = 3.141\ldots$ (Value continues)

$\pi = 3.141$ (Variable will help to fix value)

Differentiate Naming rule and Naming Convention.

→ It is used to distinguish identifier.

Naming rule: (i) It is mandatory or compulsory to follow otherwise there will be a syntax error. It is not compulsory to follow.

(ii) It helps to design identifier.

→ It is used to freeze identifier.

Naming Convention: (i) It is not mandatory or compulsory to follow.

(ii) It helps to distinguish identifier.

Container class is a class, where main method

classmate _____

is defined, then there must be one main method in a single program, to control

Date _____

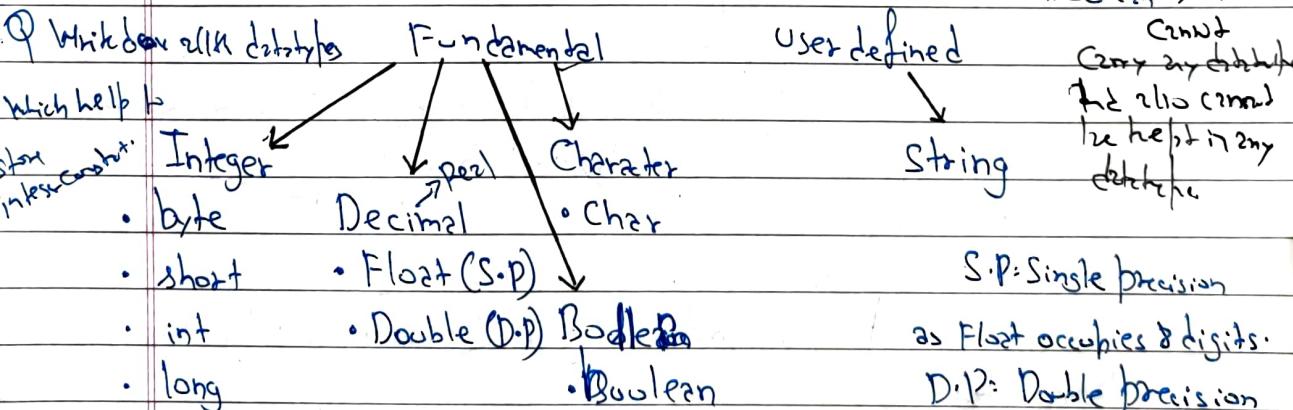
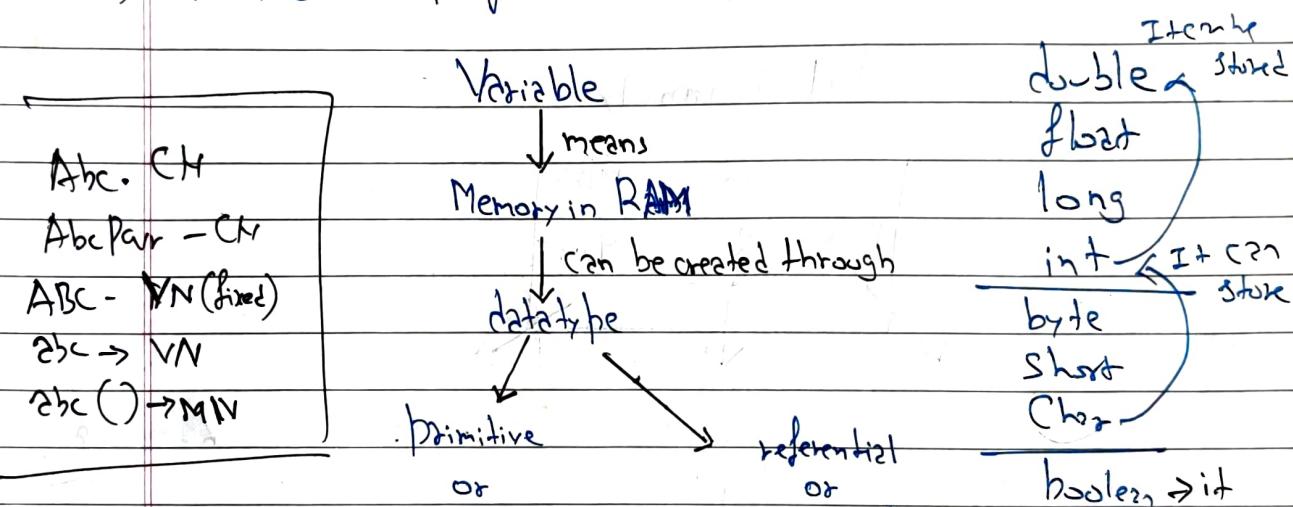
Page _____

Variable: It is a memory allocated in RAM in order to store constant.

Normally, it is created in order to deal with unknown constants. A variable is created with the help of datatype.

Datatype: Datatype is a keyword which helps to perform following two tasks:

- Helps to allocate memory by giving a name.
- Specifies the type of constant it will carry.



Memory

byte: $(-2^7 \text{ to } 2^7 - 1) = 1 \text{ byte}$

Short: $(-2^{15} \text{ to } 2^{15} - 1) = 2 \text{ bytes}$

int: $(-2^{31} \text{ to } 2^{31} - 1) = 4 \text{ bytes}$

long: $(-2^{63} \text{ to } 2^{63} - 1) = 8 \text{ bytes}$

float: 4 bytes

double: 8 bytes (predefined)

All class names are referential datatype

char: 2 bytes

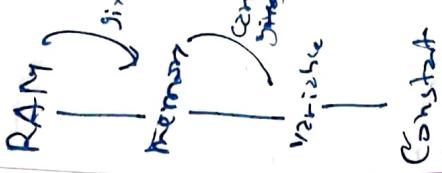
Boolean: 1 byte

String: Depends upon the number of characters. It is flexible.

Ex: BBSR: 4 characters: $2 \times 4 = 8 \text{ bytes}$

referential (starts from referential)

with address are bytes Variable



Default Integer value: int

Default decimal value: double

In Decimal when accuracy is of 4 digits then to prefer float (8 digits) no need for double precision as it will give same value by occupies more memory.

To create variable we must know the categories.

Example: Student's identity

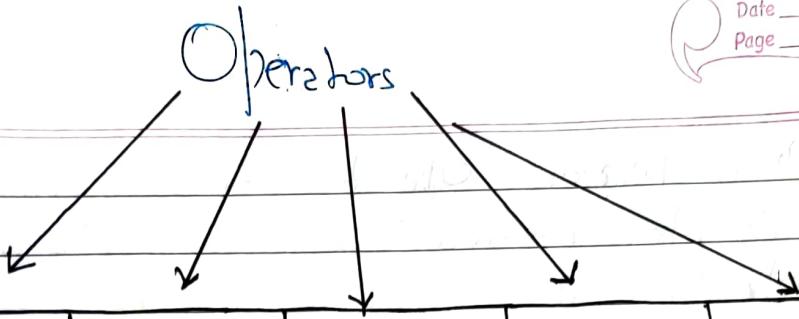
Name	Roll	Mark
PPS	30	97.5

Annotations below the table:

- The 'Name' column is labeled "String variable".
- The 'Roll' column is labeled "Under Integer. in byte as bytes (-2⁷ to 2⁷-1)".
- The 'Mark' column is labeled "Decimal (Float as accurate in 1 digit) and less memory. which will occupy less memory and storing with in other variable lead to more memory but value will be same."

According to purpose different variable are used like int, long, short.

Operator: Operators are the symbols which are used to do arithmetics and logical characters.



Mathematical	Relation	Logical	Bitwise	Shifting
$+, -, /, *$ $++, --, =, \%$	$>, <, \leq$ $\geq, ==, !=$	$\&$, $\ $, $!$		

All selection operator gives result in boolean datatype (True/False)

Two types of division (/)

(1) Integer division: The numerator and denominator are both in integer like ($4/2=2$, $0.4/2=0$) Examples.

2. Decimal division: The numerator or denominator must be in decimal but both can be when cancelled changes to integer.

($0.4/2=0.2$, $0.1/0.2=0.5$ - The decimals are cancelled but the value is in integer)

Assignment Operator: The operator helps to store the value of right side in left side variable.

$a = b * b$ Expression (b has value)
 Variable $a = 10$ constant

Module Operator: It is the operator which helps to give remainder

$$15 \% 5 = 0$$

If dividend is lower than divisor then dividend itself will be the remainder.

$$5 \% 10 = 5$$

Unary operator: Operators works on one operand.

When operator works on two operands then it is called binary operator.

Unary operators are short and notation and also

Unary operation takes only one operand. ++ is called increment operator.

It minimize the size of program.

$a++$ or $(++a)$ will

$\xrightarrow{\text{Postfix}} \xrightarrow{\text{Prefix}}$

$a++ = \text{It will increase the value of } a+1.$

$$\begin{aligned} a &= \frac{10}{a+1} \\ &= 11 \end{aligned}$$

For ex. $a = 25;$

$a++$

$S.O. \text{ print}(a);$

26

$a = 25;$

$\text{++}a;$

$S.O. \text{ print}(a);$

26

Ex-1

$\boxed{25}$ $\boxed{25}$

$a = 25;$	$a = 25;$
$b = a++;$	$b = a++;$
$S.O. \text{ print}(a)$	$S.O. \text{ print}(a)$

$\boxed{26}$ $\boxed{26}$

a b

a value is stored for one variable

Postfix: use and change = $2++, 2--$ (Example)
 Prefix: Change and use = $++2, --2$ (Example)

Unary operator = $\text{++}, \text{--}, !$

$$z = 25$$

~~S. O. P(z++);~~
~~S.O.P(z); (25)~~
~~25~~

$$z = 25$$

~~S.O.P(++z);~~
~~S.O.P(z); (26)~~
~~26~~

(--) Decrement operator: This operator help to decrease the values of a variable by one.

$z = 10;$ \rightarrow It is executed first then for more variable.
 $b = z++ + --z + z + +z;$
 $b = 10 + 10 + 10 + 11$

Short 2d notation: $++$, $--$, $+ =$, $- =$, $* =$, $/ =$, $\% =$
 \rightarrow Arithmetic assignment operators
(C.A.S.O.)

When variable are same only A.S.O is executable.

$2 +=$

Relational operators: These operation are used to giving conditions and take decision.

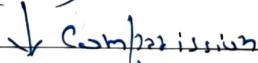
There are 6 types of relational operators.

$(=)$



1. Assignment operator or storing operator
2. It helps to store values of right side in left side variable

(\equiv)



1. Storing operator
2. It helps to compare and returns boolean type.

3. Only left side variable

3. Both side can carry numbers or variable.

$$a = 5 \quad \checkmark$$

$$5 = a \quad \times$$

Left side

on operator

$$a = 5 \quad \checkmark$$

$$5 = a \quad \checkmark$$

$$5 = 5 \quad \checkmark$$

both side
variable,
constant.

$$5 + a = 10 \quad \checkmark$$

$$a + b = = b + c$$



first add

added

then combined.

Logical operators: These are the operators which help to combine conditions.

and , or , not .

$a < b \text{ and } b > c$

It make the condition combine.

AND

Result.

A	and	B	=	
1		1	= 1	
1		0	= 0	
0		1	= 0	
0		0	= 0	

1 = True

0 = False

||: Or operator or biting symbol.

$$A \parallel B = \textcircled{O}$$

$$1 \parallel 1 = 1$$

$$0 \parallel 1 = 1$$

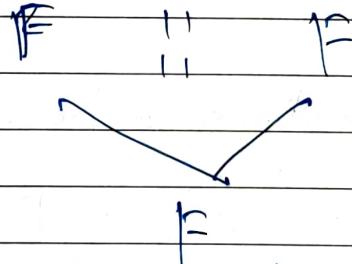
$$1 \parallel 0 = 1$$

$$0 \parallel 0 = 0$$

! : not operator (Example of relational operators)

$$!(A > B) : \text{Value} = \text{True} \quad ! = \text{False}$$

$$\begin{aligned} & a = 10, b = 10, c = 10 \\ & (a > b) \parallel !(b = c) \\ & 10 > 10 \parallel !(10 = 10) \end{aligned}$$



or operator accept
false in both sides:

Compare value of both sides of the operator. If both are not same it return true otherwise.

Order of precedence of operators: Order of precedence gives the priority to operators at the time of simplification.

Brackets

$\rightarrow () \rightarrow$ first

Unary

$\rightarrow +, -, ! \rightarrow$ second

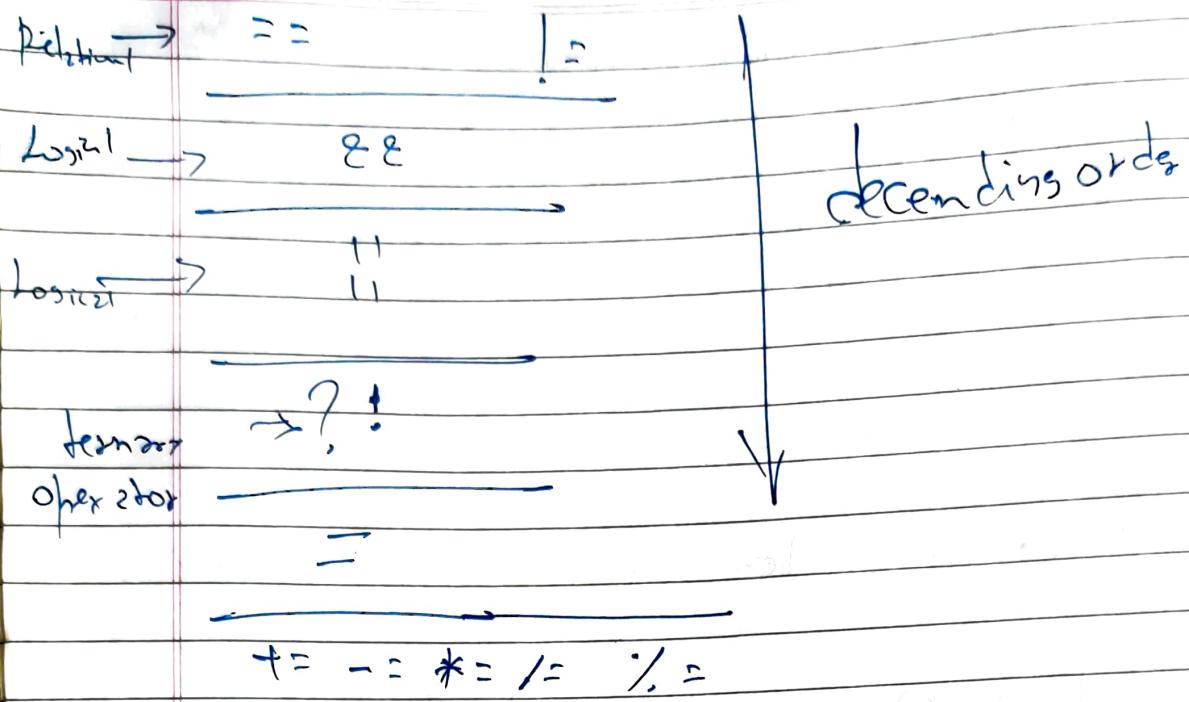
Multiplication

$\rightarrow \times, /, \% \rightarrow$ third

Relational

$\rightarrow \geq, \leq, \neq \rightarrow$ fourth

depending on de



Two types of expression.

Collection of.

Combination of operators and operators in proper sequence
is called expression.

(i) Pure expression and (ii) Mixed expression.

$4 + 3 * 2 / 7 - 6 \% 3 \rightarrow$ Pure task because
it is in pure & interse.

If same priority operators are there which ever is in the left side of the expression will get first priority.

$$4 + 6 / 7 - 6 \% 3$$

$$4 + 0 - 0$$

$$= 4$$

- (i) $3 + 2 \times 7 * 5 + 3 / 2$ 14
 (ii) $5 * (3 + (2 - 7)) * 9 / 6 - 3$ 3
 (iii) ~~$2 * 3 * 4 + 5 / 7 * 9 + 3 / 2$~~ 21

$$a = 10; b = 20;$$

$$a += b++ - --a;$$

$+=$ (Short $a +$ notation)

$$a = a + 10; \rightarrow a += 10;$$

It is an example of short $a +$ notation help to minimize the instruction size. It helps to increment the value of a variable by certain value.

→ It can be written as $a += 10;$ to get the same result

$$a = 100;$$

$$a += 10;$$

public help to make min method outside the class, user other so that method have to command to control each for available.

1. One word answer

1. It is a keyword which helps to make method universal: public
2. It is a keyword which creates a variable in order to store single precision constant: float
3. It is the object of a system class which helps to activate keyboard: in
4. It is a method which helps show message on the screen: println()

2. Identify the error.

1. int a = 225; ✓ Integer is $2^7 - 1 = 127$ but 225
2. double k = true; ✗ Double precision only accept decimal number.
3. String s = 12.5; ✗ Only in double quote.
4. boolean m = False; ✗
5. char c = 253; Only letter or symbol in double quote.
6. int d = 'A'; !

1. No error.
2. Boolean constant cannot be stored.
4. False, F is C# hi! it is a keyword
5. Integer cannot be stored.
6. No error.

public

Static

1. It determines the scope of a member outside the class

2. It can be both public, private

1. It determines the scope of a member within the objects of the same class

2. It is either public static or non static

3.

Separators: These are the symbols which are used to separate or distinguish one part of the program from another part. There are some symbols used as separators: ;, , {}, [], :, . ()

So

; (semicolon): It is called sentence terminated which helps to end a statement and allows to write further executable sentence.

Example: int a;

The statement ended with semicolon is single line sentence or instruction.

,(comma): It is a symbol used in following two ways one to separate variables at the time of creation.

Example: int a, b, c;

To separate parameters within the signature part of method.

Two values of integer type.

void add {int a, int b}
↓ parameters
↓ signature

{ } (curly bracket): This symbol helps in following ways

- To create block.
- To initialize array
Curly bracket helps to
int t; int M() = {30, 10, 20}
normal variable Array variable
- To separate elements to initialize in an array.
Block means only curly bracket.

[]

To create an array variable.

:

To create level

It helps to level a particular program or block.

A: { int, 10, }

Dot operator helps to create path.

Java.lang.*

Dot operator is also utilized as operator. (4-5) example.

()

To change the priority order within an expression. It changes the $a+b+(c-d)/e$ order of precedence. \rightarrow first priority.

It helps to design the signature part of a method prototype. Method name surrounded with parenthesis.

void add()

Use of print and println:

These are two methods which help to show the messages as well as the output on the screen.

$\text{System.out.println()}$

or

$\text{System.out.print()}$

\rightarrow only the object of the

System

long

System

long

System

long

println() is a method present in System class in order to call this method it is essential to take a help of object of system class.

System class is present in lang package.

Example $\text{System.out.println("BBSR");}$

BBSR

`System.out.print ("BBSR");`

BBSR.

Print

Println

- | | |
|--|--|
| 1. It helps to start writing from the cursor position and then the cursor will remain in same line for further printing. | 1. It helps to start writing from the cursor position sends the cursor to new line for further printing. |
|--|--|

It helps in multiple print statement.

Example

`System.out.print ("A");`

A

`System.out.print ("B");`

B

`System.out.println ("C");`

D

`System.out.println ("D");`

E F

`System.out.print ("E");`

`System.out.println ("F");`

Ex `S.O. print ("A");`

A

`S.O. p ("B");`

B

`S.O. p ("C");`

C

`S.O. p ("D");`

D

`S.O. p ("E");`

E

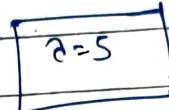
`S.O. p ("F");`

F

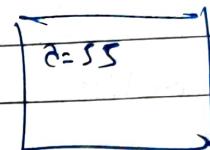
Print and Println Helps to short the variable.

To store value.

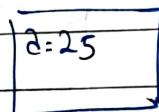
```
int a = 5;
System.out.println("a = " + a)
```



```
System.out.println("a = " + a + a);
```

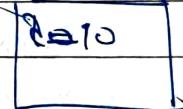


```
System.out.println("a = " + a * a);
```



```
System.out.println("a = " + a - a); // Syntax error.
```

```
System.out.println("a = " + (a + (a + a)));
```



ASCII

Unicode: (i) It stands for universal code.

(ii) It carries 65536 symbols.

(iii) Each character occupies 2 bytes.

(iv) It is used in JAVA

32 - b

48 - 0

57 - 9

65 - A

90 - Z

97 - z

122 - Z

ASCII

for
American standard code information interchange

(i) Two 256 symbols.

(ii) Each character carries 1 byte.

(iv) It is used in other high-level language.

In ASCII code is the collection of symbols

WAP to take a number then display the same number as well as the new number which will be calculated as follows
 number + unit digit of the number.

Any number divided by 10 remainder is the unit digit of the number.

/* code to display the number and a

import java.lang.*;

class A

{

public static void main (String args[])

{

int a,b,c;

a = 437;

b = a % 10;

c = a + b;

System.out.println ("The new number is" + a);

System.out.println ("The new number is" + c);

}

}

class is a fundamental block of a program

method is a fundamental block of a class

Method in a class is not possible.

2. initialize two numbers to display sum of their unit digits.
 3. name and age of a person display the name current age as well as age after fifteen years.

→ fixed

2. /* to display the sum of two numbers unit digits of the number */

import java.lang.*;

class P

{

public static void main (String args[])

{ }

int a,b,c,d,e;

a=32;

b=64;

c=a%10;

d=b%10;

e=c+d;

System.out.println("The sum of the two unit digits = "+e)

System.out.println("The sum of the two unit digits = "+(a%10+b%10))

}

3.

/* to display name, age of a person and his age after 15 years */

import java.lang.*;

class P

{

public static void main (String args [])

{
 ~~15~~
 ~~15~~

 int a,b,c;

 a = 15;

 b = 15;

 c = a+b;

 System.out.println ("PRIYAM PRATYUSHT SAHU")

 System.out.println ("The age of the boy person = " + a)

 System.out.println ("The age after fifteen years = " + c)

}

{

new keyword helps to create object.

Identify to 2 object
and produce 2 class.

Class

String p;

int a,b;

p= "PRIYAM";

a= 14;

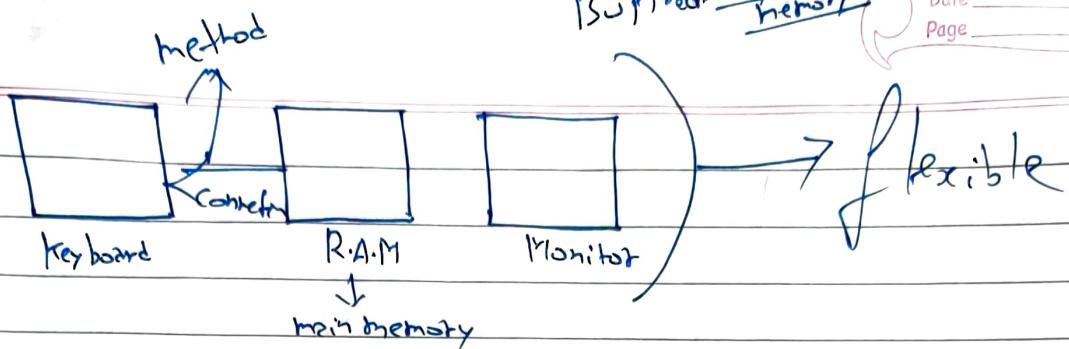
b= a+15;

System.out.println ("The name is "+s);

System.out.println ("The current age "+a);

System.out.println ("The age after fifteen years "+b);

Class helps to imagine 2 objects, where object is the
and produce 2 class.



First, active keyboard

Second, active RAM

Third, connect RAM to keyboard

To activate keyboard: (Input Stream Reader) etc \rightarrow class name
One task \leftarrow Identifier (namespace)

It helps to activate keyboard

\rightarrow coding part

To activate RAM: (BufferedReader) \rightarrow class name

Two tasks:

\leftarrow \rightarrow class

(1) Activate RAM

(2) Connect to keyboard

All classes are present in IO package.

Lang automatically activate

IO has to be active

Streams
Input Stream Reader and BufferedReader depends on
IO

*/

*/

\rightarrow free to activate all classes

input java.io.*;

\rightarrow package

Class A

i.o helps to activate any ~~base~~ class from any package

{

psvm (s.)

{

(referential variable)

keyword

new helps to create an object dynamically

(i) to allocate memory.

(ii) address to store variable

(i) Create an object

(ii) return the address of the object

activate in package

class name

Input Stream Reader

Input Stream Reader (System.in);

class referential variable

key board

object

effector

class name

(Presentation package)

System.in → will get identity through Input Stream Reader

isr → address

It helps to activate keyboard

BufferedReader br = new

BufferedReader (isr);

readLine is a method

which wait takes value from the keyboard and keep in R.A.N

^{term}
new ~~Input Stream Reader~~ (System.in);

in is the object of System class which is the representation of Keyboard.

BufferedReader br = new BufferedReader (istr);

activates
partweise
↓

Input Stream Reader
↓
BufferedReader

WAP to display a number as well as its square given by the user.

/* to display the number and its square */

import java.io.*;

Class A

{

public static void main (String args[]) throws IOException

{

InputStream

Streams

Input Stream Reader isr = new Input Stream Reader (System.in);

BufferedReader br = new BufferedReader (isr);

int a \$;

System.out.println ("enter a number: ");

Related to class name

br.readLine() (bottom word)

parseInt

$a = \underline{\text{Integer.parseInt}}(\text{br.readLine()});$

$$S = a * a;$$

`System.out.println ("The number is: " + a);`

`System.out.println ("The square is: " + S);`

}

}

10:

Method

readLine function will go to keyword to R.A.M

It is present in BufferedReader class. The value readLine takes is by default string type.

Q. Difference between `95` and `'95'`

Method

`parseInt`: which helps to convert string into integer

The `toString` is a keyword. It will ~~give~~ give the message on the monitor.

To Exception is a class which carry list of messages related to input and output keyword

~~throws~~ is a keyword to throw some appropriate error message on the screen from the exception IO exception class

1. WAP to display name and age of a person and also display age after 15 years
2. C.P. and S.P has input then calculate the profit
3. P, Rate & Time is input then calculate the simple interest

A1. /* to display name, percentage and also age after 15 years */

import java.io.*;

Class A

}

public static void main (String args []) throws IOException

{

 Stream
 InputStreamReader isr = new InputStreamReader (System.in);

 BufferedReader br = new BufferedReader (isr);

 String b;

 int a, b;

 System.out.println ("enter your name:");

 System.out.println ("enter your age:");

a = Integer.parseInt(br.readLine());
 b = Integer.parseInt(br.readLine());

$$a = b + 15;$$

System.out.println ("The name of the person: " + a);

System.out.println ("The present age of the person: " + b);

System.out.println ("The age of the person after 15 years: " + c);

}

}

A2. /* display cost price and selling price and displaying whether it is profit or not */

import java.io.*;

class P

{

public static void main (String args[]) throws IOException

{

Stream
 InputStreamReader isr = new InputStreamReader (System.in);
 Stream
 BufferedReader br = new BufferedReader (isr);

```
int a,b,c
```

System

```
a= Integer.parseInt(br.readLine());
```

```
b= Integer.parseInt(br.readLine());
```

```
c = b - a;
```

```
System.out.println ("The cost price:" + a)
```

```
System.out.println ("The selling price :" + b)
```

```
System.out.println ("The profit :" + c)
```

}

}

A3. /* display the Principle, Rate and Time and find the Simple Interest */

```
import Java.io.*;
```

Class B

{

```
public static void main (String args[]) throws IOException
```

{

```
InputStreamReader isr = new InputStreamReader (System.in);
BufferedReader br = new BufferedReader (isr);
```

Stream

```
int p,r,t,c
```

```
p= Integer.parseInt (br.readLine());
```

```
r= Integer.parseInt (br.readLine());
```

```
t= Integer.parseInt (br.readLine());
```

$$i = \frac{(p * r * t)}{100}$$

System.out.println ("The Principal is:" + p)

System.out.println ("The Rate is:" + r)

System.out.println ("The Time is:" + t)

System.out.println ("The Simple Interest:" + i)

}

}

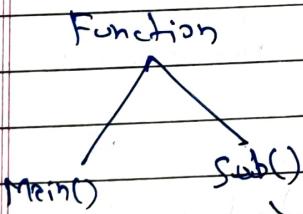
Mathematical methods

Feng

All mathematical function present in Math class.

↓
Math

Function: Function is a block which carry some instructions in order to perform certain task.



→ max()
→ min()
→ abs()
→ sqrt()
→ pow()
→ floor()
→ ceil()
→ dint()
→ rand()
→ random()
→ exp()

Sub() functions are there to help do the main function.

→ exp()

(i) Job / work of the function

(ii) Taking input, how Is it taking input, how many variables he

(iii) Is it returning output, how many, what type

For
All
Methods

If C can take 1 or more value with return, only one value.

Max(): Max function helps to get higher out of two numbers. Only two numbers.

Syntax: <datatype> variable = Math. max (a,b);

- (i) ~~<datatype>~~ = It is a single line instruction.
- (ii) Only take maximum two values of any data type. (Except Boolean Max command take positive value. String)
- (iii) It returns a value of same type or higher type as given in input.

Example: 1 Math. max (4,5); —— 5
 (40,50); —— 50
 (4,50); —— 50
 (4.0,5) —— 5.0

2. Math. max (Math. max (a,b), c)

(*) Min(): Same as Max() but in lower value

Math. min (Math. min (a,b), Math. min (c,d));

abs(): It stands for absolute - It is a method to return the same value only by removing the sign. Only take One variable. On ignore negative sign

datatype v = Math. abs (a);
 = (-3); —— 3
 (3); —— 3
 (2.5); —— 2.5

Some (1) of function of abs max().

$$A = 65$$

$$B = 66$$

$$C = 67$$

Meth. max (Math.abs(2), Math.abs(3,4));

Sqrt(): Sqrt stands for square root number given as input. It only take one value. It always returns value in double.

$$\text{double } v = \text{Math.sqrt}(2);$$

↳ any type

$$(4) \rightarrow 2.0$$

Math. remainder (10) + 4 + 'c'

$$4.0 + 4 + 67$$

$$= 75.0$$

pow(): It stands for power. It helps to determine the power. It takes only two inputs.

$$\text{double } v = \text{Math.pow}(2, 5);$$

$$10^5$$

$$(10, 5);$$

Java expression of following mathematical expression.

$$\frac{(a+bc)^2}{2ab}$$

$$\text{Math.pow}(a+b*c, 2) / (2*a*b);$$

`int v = Math.sqrt(2);`

$$\sqrt{2} \approx 1.414$$

`Math.sqrt(2 + Math.sqrt(2));`

$$\frac{(2+\sqrt{2})^{\frac{1}{2}}}{2}$$

`Math.sqrt((2+2)*2);`

2.5 - 3.2

`floor()`: It gives the lower value (2.0), (-4)

It returns the highest integer less than equal to the number to be given as input.

`ceil()`: It gives the higher value (3.0), (-3.0)

It return the lowest integer greater than equal to the number to be given as input.

`rint()`: ~~Rint~~ is a function gives rounded function. (2.0), (3.0)
rint returns the rounded integer in integer form. If number is double. It gives previous value if in 0.5.

`round()`: `int v = Math.round(2);`

$$4.5 = 5$$

It gives rounded integer in integer form.

$$2.3 = 2$$

It gives next value if in 0.5

If value is given in double it gives value in long.

And in other case it gives value in int.

int and round differ in 0.5

Exp default value is e

e^s

`exp()`: It stands for exponent. where the base is e
Example:

`Math.exp(5);`

$(2 \times e^5)^7$

`Math.pow(2 * e * Math.exp(5), 7);`

`random()`: It is a method it returns value in double which is in between 0 and 1. It doesn't take any value. (parameter). Excluding both the end. It returns value in double.

`Math.random()`

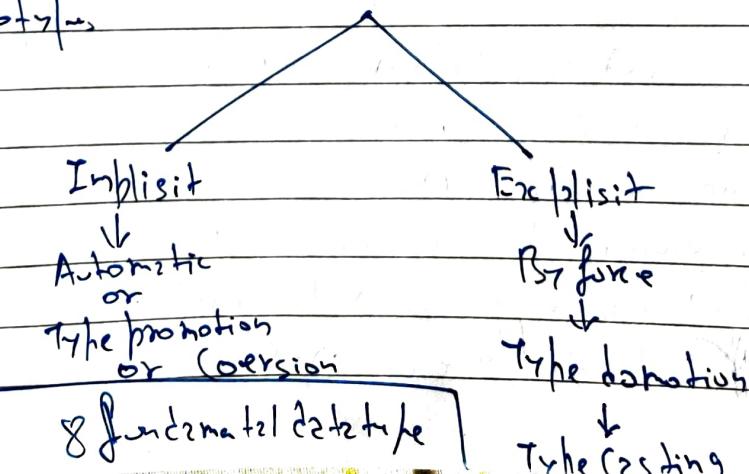
0.33, 0.25

Type Conversion

Type conversion means fundamental datatype conversion.
It is a process of converting values of one datatype into another fundamental datatype.

It is the rule of the conversion which help to perform this conversion.

Typecasts



Implicit: (i) It is a type of conversion which takes place automatically by the compiler.

(ii) It is conversion from lower data type value to higher data type value.

Explicit: (i) It is a type of conversion which will done by programmer through instruction.

(ii) It is a type of conversion it takes place higher data type value is to be stored in lower data type variable.

int p (int) → type casting

Math :: pow(1,2)

(promising
float)

We have to convert

int p = "A"; ✓ → Automatic conversion

int char p = 65; ✗

char p = (char) 65; ✓ → Explicit conversion.

$$\frac{1}{2} = 0$$

$$(\text{double}) \frac{1}{2} = 0.5$$

Use of ternary operator or conditional operator (?:)

This operator helps to take decision and perform a specific task.
 It is called ternary operator as it works on three operands.
 Syntax of conditional operator is as follows:

$y = (\text{condition})? E1: E2;$



Example:

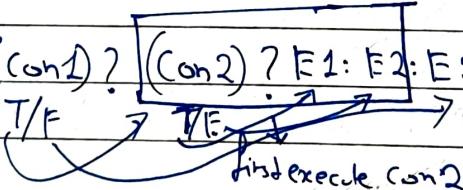
- (i) $a=10, b=5$
- (ii) $a=10, b=10$
- (iii) $a=5, b=10$

$y = (a > b)? a: b;$

Values $y =$

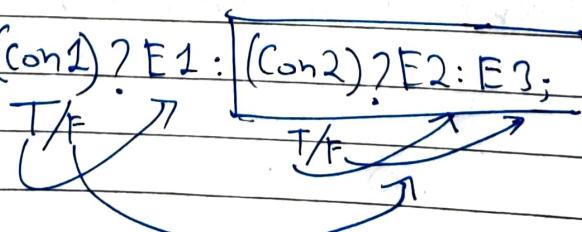
- (i) $y = 10$
- (ii) $y = 10$
- (iii) $y = 10$

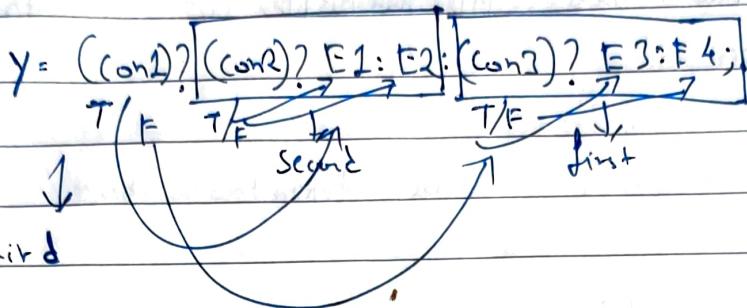
$y = (\text{con1})? \boxed{(\text{con2})? E1: E2: E3;}$



Then combined with Con1 at statement y.

$y = (\text{con1})? E1: \boxed{(\text{con2})? E2: E3;}$





1.

$$m = ((a+b) \times s)(a+b \times s) ? \quad a+10 : \quad (b)_! = (c+s) ? \quad b+s : c+s ;$$

$$\Rightarrow m = (5+s \times 5) ? \quad a+10 : \quad [(5)_! = 3+s] ? \quad 5+s : 3+s ;$$

$$\begin{cases} a=5 \\ b=5 \\ c=3 \end{cases}$$

$$m = (10 \times 5) ? \quad 15 : \quad \notin 10$$

WAP to take two numbers as input and display both higher and lower number.

/* display the higher and lower value by taking two number */

import java. ~~io~~. *;

Class P

{

public static void main (String args[]) throws IOException

{

BufferedReader br = new BufferedReader (new InputStreamReader (System.in))

Stream

int a, b;

System.out.println ("Enter any two numbers");

a = Integer.parseInt (br.readLine());

b = Integer.parseInt (br.readLine());

System.out.println ("Higher no is: " + Math.max (a, b));

System.out.println ("Lower no is: " + Math.min (a, b));

}

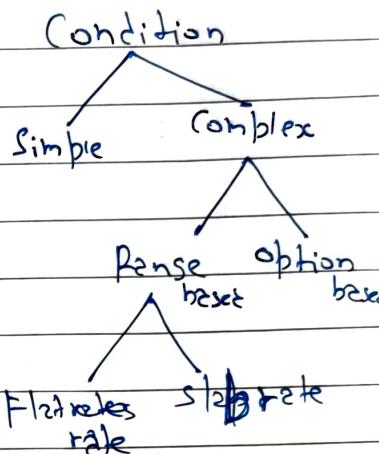
}

Uses of conditional statements

Conditional statements are used in a program in order to take decision.

Three types of conditional statements.

- (i) If if else
- (ii) if else if else
- (iii) switch case

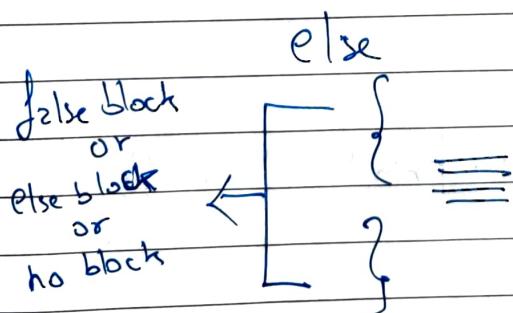
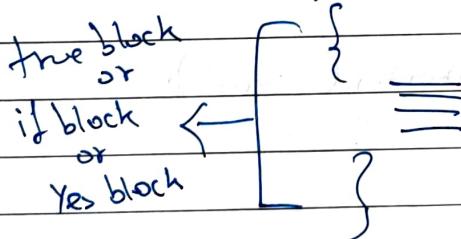


Simple: ~~work on~~ work on if, else

Use of if else.

It is combined keyword which is used to give condition in a program.

↪ Syntax: `if (condition)`



If condition is true
then if block will be
activated, otherwise
the control will
move to else.