OOPs

- Object oriented Programming System.
- class & object ಆಧಾರಿತ program Design ಮಾಡುವ ಭಾಷೆಗಳು OOPs. , [C++, JAVA, PYTHON → USED LANGUAGES]

Advantages

- Code Reusability through Inheritance.
- Security through Encapsulation
- Flexibility through Polymorphism.
- Modularity. (seperate classes ಆಗಿ divide ಮಾಡಬಹುದು).
- PROBLEM SOLVING (BREAK DOWN THE PROBLEMS).

⇒ Difference Both POPs & OOPs

POP (Procedure Oriented Programming)	OOPs (Object oriented Programming).
• Based on Functions.	• Based on classes & objects.
• Top-down Approach	• Bottom up Approach.
• High Complexity (problem solving)	• 4 pillars.
	• Data security
	• Data Hiding
Eg: C language	Eg: Java, C++.

CLASS

- Variables, methods, and constructors ഒരുമിച്ചു define ചെയ്യാൻ ഉപയോഗിക്കുന്ന Logical Form.
- class ന് memory space ഉണ്ടാവില്ല.
-
-

OBJECT

- Instance of class -
- Object ന് memory space seperate ആയിരിക്കും (ഉണ്ടാകും).
- class ലെ methods ഉം Variables ഉം Access ചെയ്യുന്നത് object വഴിയാണ്.

  [Note: Static വഴിയും Access ചെയ്യാം (object ഇല്ലാതെ).

new Keyword

- object ന് memory space allocate ചെയ്യാൻ ഉപയോഗിക്കുന്നു.

CONSTRUCTOR.

- object ന Initialize ചെയ്യാൻ ഉപയോഗിക്കുന്ന Special method ആണ്.
- class ന്റെ same name ആയിരിക്കും.
- Return type ഉണ്ടാവില്ല.

Note : [override ചെയ്യാൻ പറ്റില്ല. But overload ചെയ്യാം].
(Base class hide ആയിരിക്കും).

Additional.
- ~~object create ചെയ്യുന്ന സമയത്ത് മാത്രം ca~~

TYPES OF CONSTRUCTORS.

1) Default
2) Non Argument
3) Parameterized.

Note : (4) COPY CONSTRUCTOR.

1) Default Constructor.

- class create ചെയ്യുമ്പോൾ നമ്മൾ Default ആയി Compiler constructor create ആകും.
- Jvm automatically create ചെയ്യുന്ന constructor ആണ് Default constructor.

2) Non Argument / Non Parameterized.

- Constructor ന്റെ ഉള്ളിൽ Argument pass ചെയ്യുന്നില്ലെങ്കിൽ അതാണ് non Argument ആണ്.

3) Parameterized : Argument pass ചെയ്യുന്നുണ്ടെങ്കിൽ, ie parameterized

Note:

Argument vs Parameter Differences

Additional

4) copy constructor:

- object ന pass ചെയ്യാൻ വേണ്ടിയാണ് copy constructor.

Static Keyword.

- static ൻ വേണ്ടി നമ്മൾ static method ഉപയോഗിച്ച് static variables access ചെയ്യാൻ പറ്റും.
- static ൻ object create ചെയ്യേണ്ട ആവശ്യമില്ല.
- static common memory space ആണ് നൽകുന്നത്.
- It is a non access modifier.

- Refer non Access modifiers, Access modifiers. →.

4 Pillars (5)

1. Inheritance
2. Polymorphism.
3. Encapsulation.
4. Abstraction.

⇒ INHERITANCE

- ഒരു class ൽ നിന്ന് മറ്റൊരു class ലേക്ക് variable ഉം methods ഉം Access ചെയ്യുന്ന രീതിയാണ് Inheritance
- Code Reusability.
-

TYPES.

1. Single Inheritance
2. Multilevel Inheritance
3. Hierarchical Inheritance.
4. Multiple Inheritance.
5. Hybrid Inheritance

Refer → syntax + Eg images.

Note: • Multiple Inheritance not supported in Java. (6)

⇒ By using interface ⟶ we use multiple Inheritance (support)

⇒ POLYMORPHISM.

- ഒരു method ന്റെ different രീതിയിൽ use ചെയ്യുന്നതിനെയാണ് polymorphism.

- TYPES OF POLYMORPHISM.

1) compile-time Polymorphism (static / Early Binding)

Eg: Method overloading

2) Runtime Polymorphism (Dynamic / Late Binding).

Eg: Method overriding.

⇒ Method overloading

- Same method ന്റെ പേര് different രീതിയിൽ use ചെയ്യാൻ പറ്റും.

  - Different parameters.
  - Order of parameters different.
  - Number of parameters different (count).

⇒ Method Overriding

- Base class ലെ ഒരു Sub class ലെ ഒരു method ഉം Same രൂപത്തിൽ Base class ലെ method നെ hide ചെയ്യും.
- here, same method with same parameter list.
- Base class നെ access ചെയ്യണമെങ്കിൽ super keyword use ചെയ്യണം
- super keyword point ചെയ്യുന്നത് തൊട്ടു മുകളിലെ class നെയാണ് (Base class).

⇒ OPERATOR OVERLOADING.

- ഒരു operator നെ Different രൂപങ്ങൾക്ക് use ചെയ്യാൻ സാധിക്കുന്നതാണ് Operator Overloading.

eg : +

(3) ENCAPSULATION

→ ഒരു Method അല്ലെങ്കിൽ Variable അല്ലെങ്കിൽ ഏതൊരു Single Unit ആക്കി ചെയ്യുന്നതാണ് Encapsulation.

→ Data hiding

→ ഒരു Method നെ നമ്മൾ private ആക്കിയിട്ട് declare cheyyanam.

→ Private ആക്കിയ method നെ നമ്മൾ getter Setter വഴി നമ്മുടെ Access ചെയ്യാം. (Modify cheyyaam)

→ Developer ന് മാത്രമേ Accen ചെയ്യാൻ പറ്റൂ.

→ Not Accen → by using this Keyword.

(4) ABSTRACTION

- Implementation details hide ചെയ്ത് കൊണ്ട് functionality മാത്രം user ന് provide ചെയ്യുക.

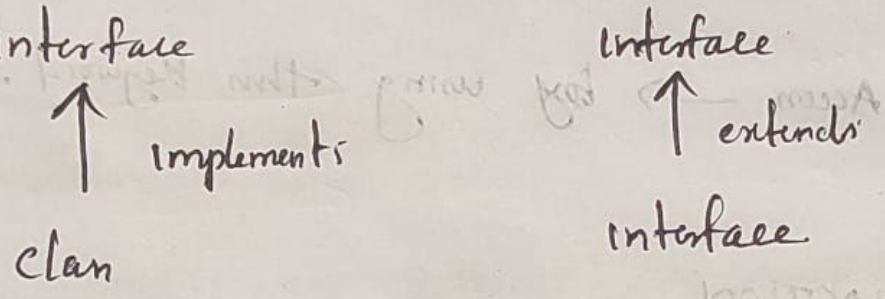
  eg: Scanner class, println().

- Abstraction can be achieved by 2 ways.
  1) Abstract class (0 to 100%)
  2) Interface (100%)

Abstract class

- abstract class ന object create ചെയ്യാൻ പറ്റില്ല.
- abstract class ൽ body ഉണ്ടായിരിക്കും.
- abstract class ൽ abstract methods ഉം non abstract methods (non access modifier).

Interface

- implements keyword use ചെയ്യും
- body ഉണ്ടായിരിക്കില്ല, object create ചെയ്യാൻ പറ്റില്ല.
- 

Exception Handling

Exception

- runtime ൽ വരുന്ന (user invalid input) or user ചെയ്യുന്ന തെറ്റായ വരുന്ന mistakes handle ചെയ്യുന്നതാണ് exception.
-

try

- exception code (വരാൻ സാധ്യതയുള്ള error കൾ) try block ൽ ചെയ്യണം.

catch (multiple catch block ഉപയോഗിക്കാം).

- exception handle ചെയ്യുന്നതാണ് catch ഉപയോഗിക്കുന്നത്.
- exception object create ചെയ്ത് ആ object ൽ, ഉണ്ടാവാൻ സാധ്യതയുള്ള exceptions display ചെയ്യിക്കുക.
- try must follow catch.
- try ഉള്ളതായ catch മാത്രം ഉണ്ടാകാൻ പാടില്ല, But catch ആദ്യം ഉണ്ടാകാൻ കഴിയും.

finally

- must ആയിട്ടും work ചെയ്യേണ്ട code finally block ൽ ചെയ്യാം (It is executed exception handled or not).

throw

- Developers തന്നെ create ചെയ്യുന്ന exception throw ൽ ഉപയോഗിക്കുക.

throws

- Any exception ൽ, warning ഒഴിവാക്കുന്നതാണ് throws.

System . out . println ( ) → . operator use ചെയ്ത് access ചെയ്യുന്നു.

↙ ↓ ↘

clan object method.

finalize ()

- unwanted ആയിട്ടുള്ള System Resource ഒക്കെ Remove ചെയ്യാനാണ് use ചെയ്യുന്നത്.
- memory free ആക്കാൻ വേണ്ടിയാണ് finalize().

final Keyword

Variable ⟶ To create constant variable

method ⟶ override ചെയ്യാൻ പറ്റില്ല.

clan ⟶ Inherit ചെയ്യാൻ പറ്റില്ല.

this Keyword

- Global Variable ഉം access ചെയ്യാൻ വേണ്ടിയാണ് (Same name ഉള്ള variables ഉം)
- Same name ഉള്ള variable ഉം Globally access ചെയ്യാൻ വേണ്ടിയാണ് this Keyword.

Super Keyword.

polymorphism ലൂടെ method overriding ചെയ്യുമ്പോൾ Base class ന്റെ hide ചെയ്തിട്ട് subclass work ചെയ്യാം.

A Base clan ലുള്ള method നെ access ചെയ്യാൻ ചിലപ്പോൾ super keyword use ചെയ്യുന്നത്.

Note: [നമുക്ക് മുകളിലുള്ള Base clan നെ മാത്രമേ Access ചെയ്യാൻ പറ്റൂ].

A, B, C 3 clan ഉണ്ടെങ്കിൽ (inherited class)

A clan ഉം B clan ഉം Inherit ചെയ്ത്

B clan ൽ super keyword വിളിക്കുക.

A clan നു access കിട്ടും. (method നു മുകളിലാണ് super keyword വിളിക്കുന്നത്)

Same as C.

Keyword ↑ method name ↑

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

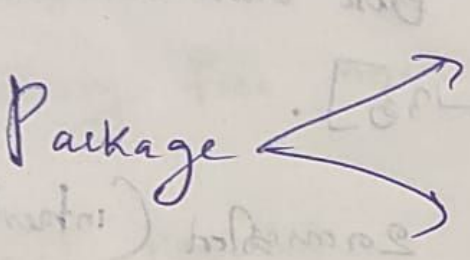
public ↓ Access modifier

void ↓ Return type

String args[] ↓ array of String type

Package.

- used to group related classes
- like folder in a File directory
- we use packages to avoid name conflicts.



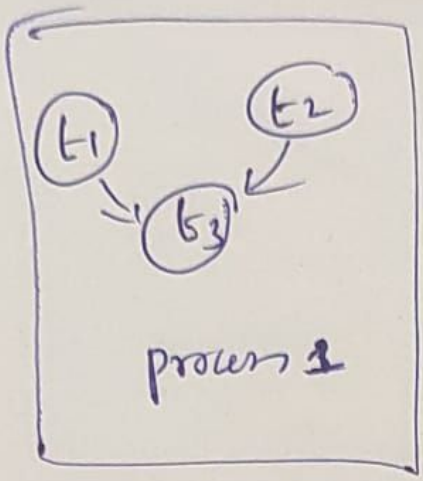
- Import Keyword
  - To use a class or package from the library

Syntax: `import package.name.*;`

Example:

```
import java.util.Scanner;
```

Thread ..



- sharing same memory (14)

- Every program has one thread.
- perform multiple things..
- used to perform ~~multiple things~~ complex operations without any disturbance in the main program.

Multi Threading.

- Various multiple Threads are executed at the same time