

A1

1. Classification of programming languages

1.1 Imperative (ภาษาเชิงคำสั่ง) → ทำให้คอมพิวเตอร์ประมวลผลอย่าง "เป็นลำดับขั้นตอน"

↳ ลักษณะการประมวลผลมีผลต่อประสิทธิภาพการทำงาน

Von Neumann : จัดให้เป็นโปรแกรมเป็นข้อมูลประเภทหนึ่ง เช่น Fortran, Pascal, C

Scripting : เขียนสั้น+ง่าย เกิดจากการเรียงคำสั่งที่มีมาก่อนอยู่แล้ว

สามารถนำไปประมวลผลกับสภาพแวดล้อมที่เฉพาะในลักษณะของ Interpreter เช่น Python, Ruby, JavaScript, PHP

รวมถึง dynamic general-purpose language

Object-oriented : ส่วนหลักจะอาศัย Class หรือ Object ซึ่งบรรจุ Method ไว้ภายใน

↳ ความสัมพันธ์ระหว่าง class เป็นแบบ semi-independent (อิสระ แต่ก็มีความสัมพันธ์กันอยู่) เช่น C++, JAVA

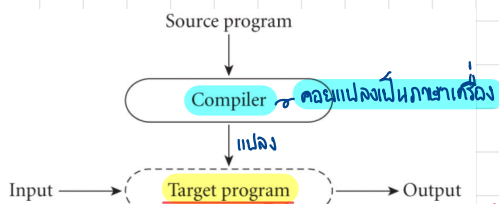
1.2 Declarative (ภาษาเชิงประกาศ) → ไม่ได้กำหนดลำดับขั้นตอน แต่จะกำหนดคุณสมบัติของคำตอบของปัญหา

Functional : Based on recursive definition of functions

Program is a function from inputs to outputs เช่น Lisp, Scheme, ML, Haskell

Logic : บอกคุณสมบัติหรือแก้ปัญหาในรูปแบบของ rule เช่น Prolog

2. Program Translation : Compiler



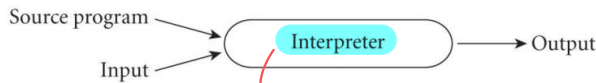
ข้อดี : โปรแกรมเร็ว (เพราะการตัดสินใจบางอย่างทำตั้งแต่ขั้น Compile เช่น การกำหนด memory address ของตัวแปร)

ข้อเสีย : ไม่ Flexible (ต่างเครื่อง ก็ต่างกัน)

มีไว้ใน target program ได้เลย ที่ขั้นตอนการทำงานระบุตำแหน่งไว้ชัดเจน

↳ สามารถนำไปใช้คอมพิวเตอร์ประมวลผลได้ "โดยไม่ต้องทำการแก้ไขใหม่" ยกเว้นว่าโค้ดจะมีการเปลี่ยนแปลง

3. Program Translation : Interpreter



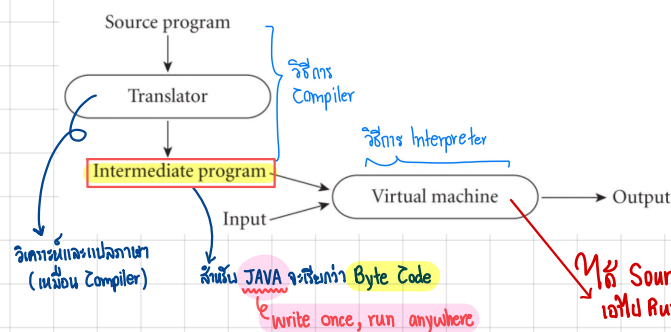
ข้อดี : แก้ไขข้อผิดพลาดง่าย + Flexible

ข้อเสีย : ช้า (ถ้าเป็น loop จะช้ามาก)

วิเคราะห์ว่า Source Code ถูกตีความหลักภาษาไหน
ถ้าต้องการเปลี่ยนเป็นภาษาเครื่องก็ทำตามคำสั่งต่อไป
ที่ภาษานั้น แม้ว่า Code จะไม่ผิดแต่ก็
ทำแบบ "ทีละบรรทัด"

หมายเหตุ : ไม่ได้เปลี่ยนเป็นภาษาเครื่องโดยตรง แต่จะเตรียม function ต่างๆ
ในรูปของภาษาเครื่องไว้ก่อนแล้ว และจะเรียงฟังก์ชันเหล่านี้เพื่อแปลงภาษาแทน

4. Program Translation : Compiler and Interpreter



วิเคราะห์และแปลภาษา
(เหมือน Compiler)

สำหรับ JAVA จะเรียกว่า Byte Code

write once, run anywhere

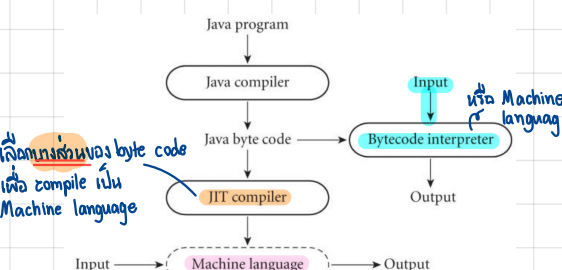
ได้ Source code เดียว ที่สามารถ
เอาไป Run กับเครื่องอื่นได้ (ไม่ต้อง Compile ใหม่)

ไม่ต้อง compile ใหม่ / ไม่ต้องเขียนโปรแกรมเพิ่ม

ข้อดี : เคลื่อนย้ายได้ง่าย (สามารถนำไปทำงาน platform ใดก็ตามที่มี Virtual Machine ติดตั้งอยู่)

ข้อเสีย : ขั้นตอนการแปลในทุกข้อความสั่ง (ช้า)

5. Just-in-Time Translation



เลือกบางส่วนของ byte code
เพื่อ compile เป็น
Machine language

ทำให้ทำได้โดยอัตโนมัติ ไม่ต้องทำขั้นตอนแปล

คำสั่งที่ถูกเรียกใช้บ่อยจะถูก Compile เป็น Machine Language แล้วเก็บไว้ในหน่วย

คำสั่งที่ไม่ถูกเรียกใช้บ่อยจะถูกกระทำผ่าน interpreter

มักพบใน modern language

A2

Binding = การจับคู่นามหรือสิ่งที่ขึ้นกับสิ่งที่ยังไม่ถึง

Binding time = time at which an association is created

decision is made) ภาษาส่วนใหญ่เป็นแบบ static เช่น C, C++, JAVA

Static binding = Things are bound before run time (early binding) → อาจไม่ flexible เพราะเปลี่ยน data type

Associated with greater efficiency, e.g. compiler decides on layout of variables in memory and generates efficient code to access them.

Compiled languages tend to have early binding times. (มักจะมี Run time หรือตอนรันเลย)

Dynamic binding = Things are bound at run time (late binding)

Associated with greater flexibility, e.g. decision on which data value of which type is bound to a variable

name may be made at run time → ใช้! เวลาเปลี่ยนประเภท data type

Interpreted languages tend to have later binding times. → ช้า!

ภาษาที่ไม่มี compiler

Object Lifetime and Storage Management

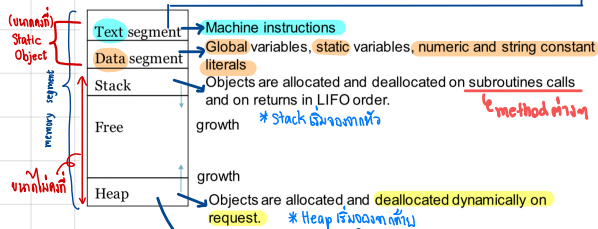
Lifetime is time between creation and destruction.

An object to which a name is bound has its object lifetime.

Object lifetime corresponds to its storage allocation.

เก็บ code ที่แปลงเป็น machine code (Source code)

หรือบางทีก็เป็น โค้ดที่เครื่องเข้าใจ



Fixed size. Static objects are given an absolute address and retained throughout program execution.

พอสั่ง method ชื่อ f1
f1 } ปิดท้าย
main }
2

ส่วนของ memory ที่ใช้เก็บ data ที่เราต้องมีการ "สั่งจอง" ด้วยคำสั่งเฉพาะ เช่น new ใน JAVA (ส่วนมากเป็น pointer)

เนื่องจากข้อนี้โจทย์ถามว่า switch can use range นั่นก็หมายความว่าเราสามารถเขียน case เป็นช่วงได้ ดังนั้นข้อนี้ควรใช้ switch เพราะ switch จะมีการสร้าง jump table ทำให้เมื่อเราป้อนข้อมูลเข้าไป จะกระโดดไปทำตามเงื่อนไขที่ไม่ต้องเช็คทีละครั้งแบบ if-else)

STATIC VS DYNAMIC METHOD

(JAVA)

S.N.

Static Binding

Dynamic Binding

1. A type of polymorphism that collects the information to call a method during the compile-time.
2. The binding happens at compile time.
3. The actual object is not used for binding.
4. It is also called early binding because binding happens during compilation.
5. The execution speed is high.
6. Method overloading is the best example of static binding.
7. The methods which are private, static and final, show static binding because we can not override them.

1. A type of polymorphism that collects the information to call a method at the runtime.
2. The binding happens at runtime.
3. The actual object is used for binding. *Object จริง! ที่ new มา*
4. It is also called late binding because binding happens at run time.
5. The execution speed is slow.
6. Method overriding is the best example of dynamic binding.
7. The methods other private, static and final methods show dynamic binding because overriding is possible with these methods. *เช่นใน c++ ที่ใช้ Virtual*

Code to understand the static binding in Java

```
1. package com.techvidvan.binding;
2. class Person
3. {
4.     public void speak()
5.     {
6.         System.out.println("Person speaks");
7.     }
8. }
9. class Teacher extends Person
10. {
11.     public static void speak() → override ไม่ :: Static method
12.     {
13.         System.out.println("Teacher speaks");
14.     }
15. }
16. public class StaticBinding
17. {
18.     public static void main( String args[ ])
19.     {
20.         // Reference is of Person type and object is Teacher type
21.         Person obj = new Teacher(); object
22.         obj.speak();
23.         //Reference and object both are of Person type.
24.         Person obj2 = new Person();
25.         obj2.speak();
26.     }
27. }
```

Output:

```
Person speaks
Person speaks
```

Code to understand the Dynamic Binding in Java:

```
1. package com.techvidvan.binding
2. class Person
3. {
4.     public void speak()
5.     {
6.         System.out.println("Person speaks");
7.     }
8. }
9. class Teacher extends Person
10. {
11.     @Override
12.     public void speak()
13.     {
14.         System.out.println("Teacher speaks");
15.     }
16. }
17. public class DynamicBinding
18. {
19.     public static void main( String args[])
20.     {
21.         //Reference and objects are of Person type.
22.         Person obj2 = new Person();
23.         obj2.speak();
24.         // Reference is of Person type and object is Teacher type
25.         Person obj = new Teacher();
26.         obj.speak();
27.     }
28. }
```

Output:

```
Person speaks
Teacher speaks
```

Type Clash?

```
System.out.println(2 + 3 + ">=" + 1 + 1);
```

5 >= 2 us

expressions are evaluated from left to right, in this case 2 + 3 get summed to 5 and when "added" to a string result in "5 >=", which when added to 1 gives "5 >= 1", add another 1 and your result is: "5 >= 11"



If-else Vs Switch

If-else		switch
Definition	Depending on the condition in the 'if' statement, 'if' and 'else' blocks are executed.	The user will decide which statement is to be executed.
Expression	It contains either logical or equality expression.	It contains a single expression which can be either a character or integer variable. (เป็น logical ไม่ได้)
Evaluation	It evaluates all types of data, such as integer, floating-point, character or Boolean.	It evaluates either an integer, or character.
Sequence of execution	First, the condition is checked. If the condition is true then 'if' block is executed otherwise 'else' block	It executes one case after another till the break keyword is not found, or the default statement is executed.
Default execution	If the condition is not true, then by default, else block will be executed.	If the value does not match with any case, then by default, default statement is executed.
Editing	Editing is not easy in the 'if-else' statement.	Cases in a switch statement are easy to maintain and modify. Therefore, we can say that the removal or editing of any case will not interrupt the execution of other cases.
Speed	If there are multiple choices implemented through 'if-else', then the speed of the execution will be slow.	If we have multiple choices then the switch statement is the best option as the speed of the execution will be much higher than 'if-else'.

```
switch (input) {
  case "java":
    System.out.println("Java Current Version is 12.");
    break;
  case "python":
    System.out.println("Python Current Version is 3.7");
    break;
  case "rust":
    System.out.println("Rust Current Version is 1.34.1");
    break;
  default:
    System.out.println("We don't have information about th
}
}
```

```
int input = 15 ;
switch (input) {
  case input >= 30 :
    System.out.println("yey") ;
}
```

นั่นเอง ซึ่งจะเห็นว่าจะใช้ค่าที่สนใจเพื่อเลือกที่อยู่ที่จะกระโดดไปทำงาน ณ บรรทัดที่ต้องการได้เลย โดยไม่ต้องคอยตรวจสอบทีละ case ว่าค่าที่สนใจตรงกับเงื่อนไขหรือเปล่าทำให้สามารถทำงานได้เร็วกว่าเมื่อเทียบกับ nested if

```
int input = 15 ;
switch (input) {
  case 15 :
    System.out.println("yey") ;
  case 16:
    System.out.println("yey2") ;
  case 17:
    System.out.println("yey3") ;
}
```

ถ้าไม่มี break จะแสดงผลลัพท์ออกมาหมดเลย (Fall through)