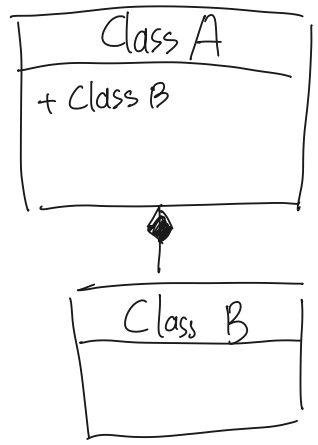
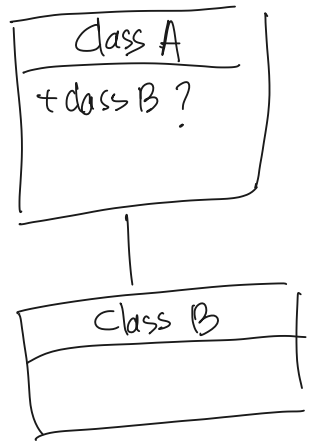
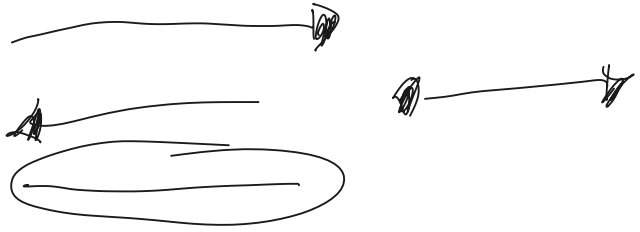
Composition



Aggregation



Assoation ⇒ enum



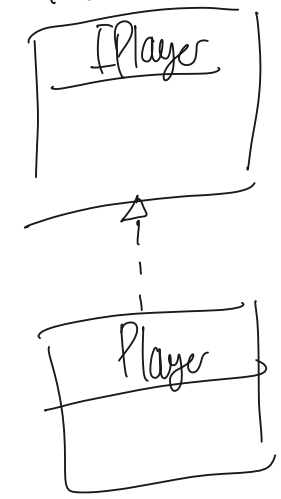
```
class GameController
{   Board board = new Board();
}
```

Composition

```
class GameController
{   Board board;
    public GameController(Board b)
    {   board = b;
    }
}
```

Dependency

Realization



Program.cs

Piece[,] piece = game.GetPieceOnBoard();

C.W( )

8×8×12 = 600-700

Class GameController

+ Action<Piece, Position> action

+ GetPieceOnBoard() : Piece[8,8]

+ MovePiece( )

{

action.Invoke(Piece, Post)

}

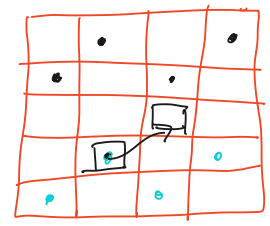
+ Action<Piece, Position, Position>

UI.cs

+ void UpdateUI(Piece, Position)

C.W( ) → Piece = 3

→ 2,2



Piece[8,8] p = GetPieceOnBoard();

p[2,2] = null

p[3,2] =

Program.cs

+ OnPieceUpdate(Piece, Position, Piece)

{

for(_ , _ , _)

for(_ , _ , _)

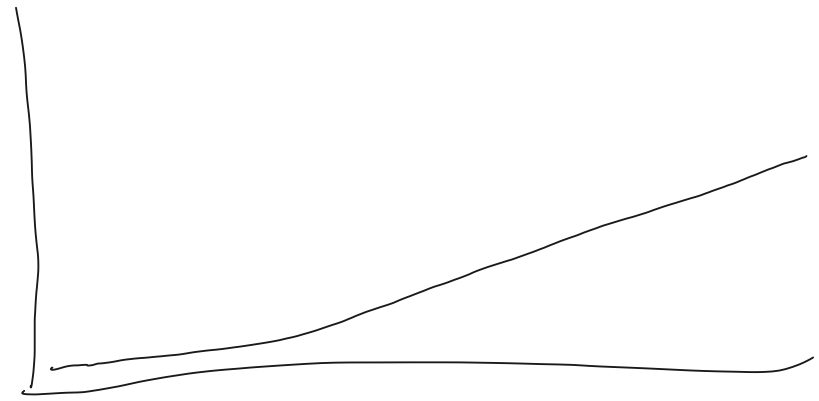
p = Piece.id;

}

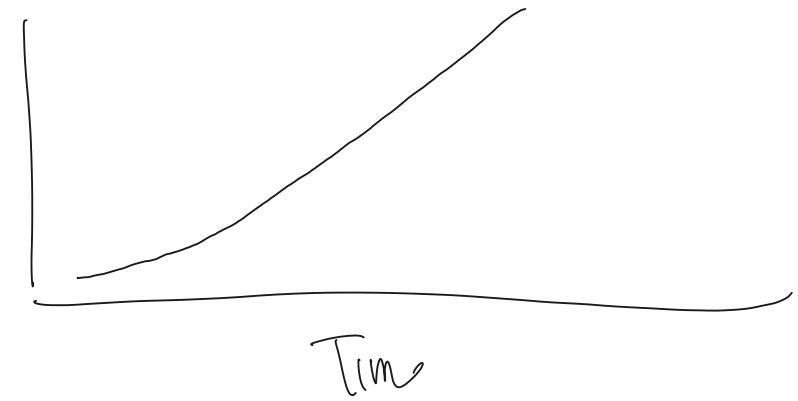
p 3

C / C++	C#
Low Level (Near assembly)	High Level (Human Language)
Functional Programming	OOP
Manual Allocation Manual Free	Garbage Collection
Performance	–
–	Learning Curve easier
Single platform	Multi platform
For small device	For faster / complex device

C / C++



C #

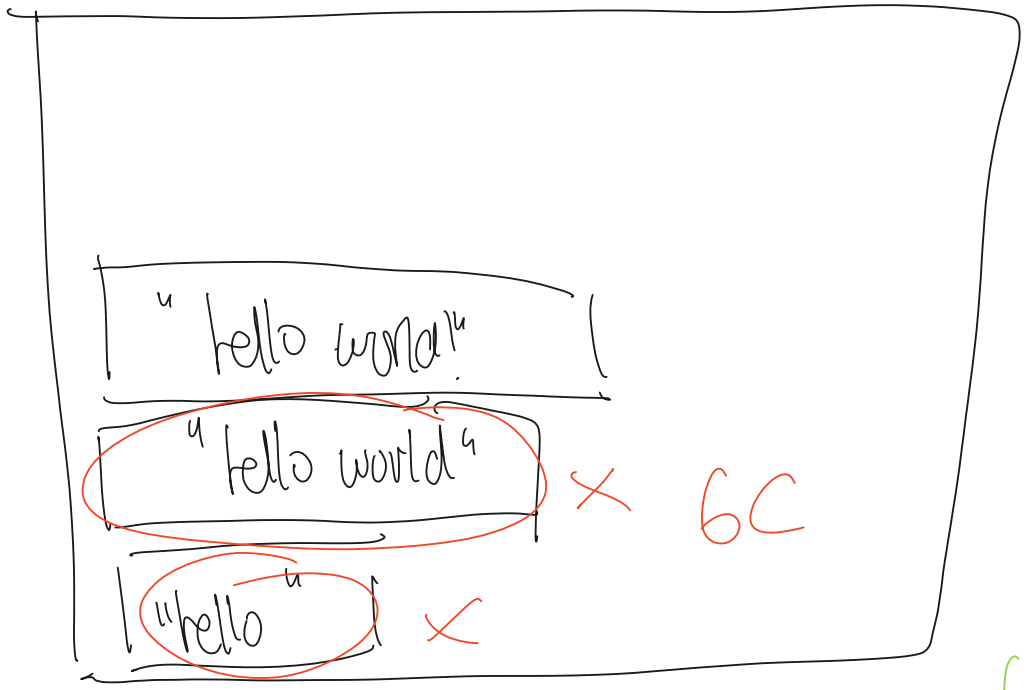


Garbage Collection

Garbage Collector

Factor
- how much garbage
- memory full (near)
- time from last Collection

managed heap / managed resource
- class
- string
- array
- collection
- dicb.



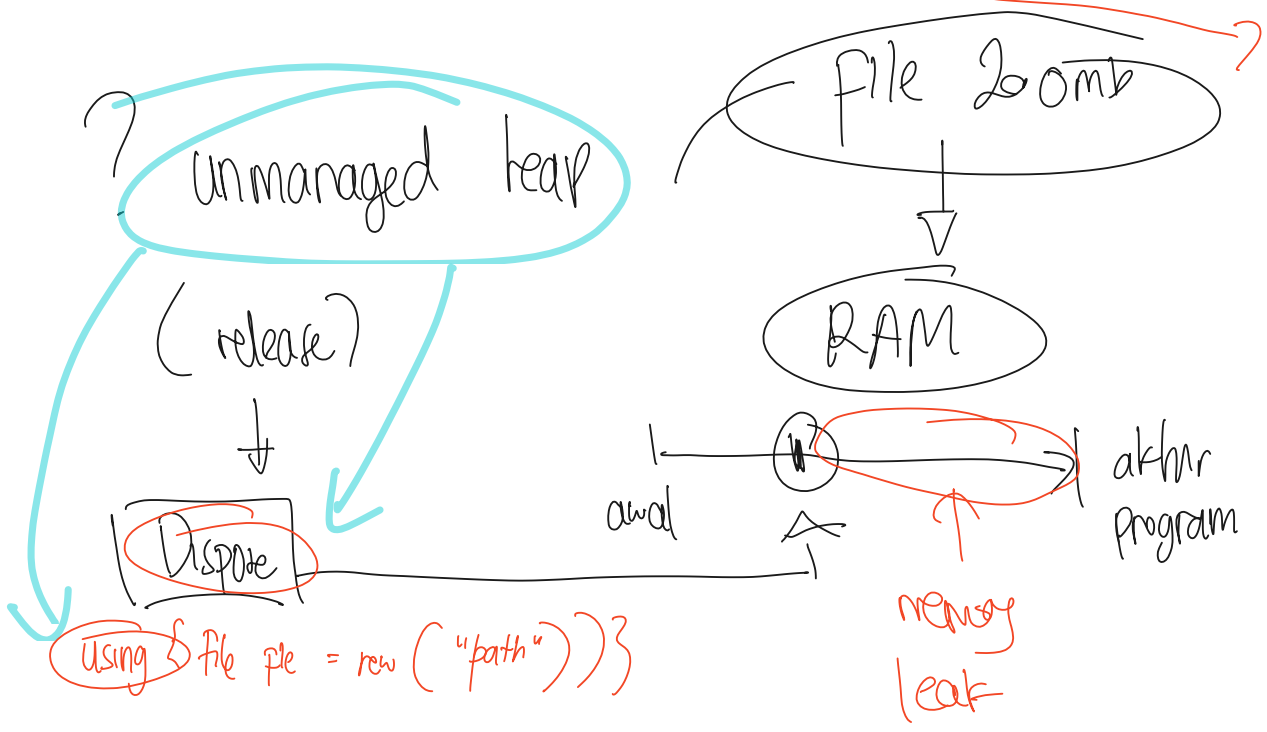
memory
- stack ✗
- heap
  - managed heap (array, class, string, dcb)
    - (internal Program)
  - unmanaged heap (external) ✗
    - (file, api, http request, database, smtp)

GC → Managed heap

60

61+60

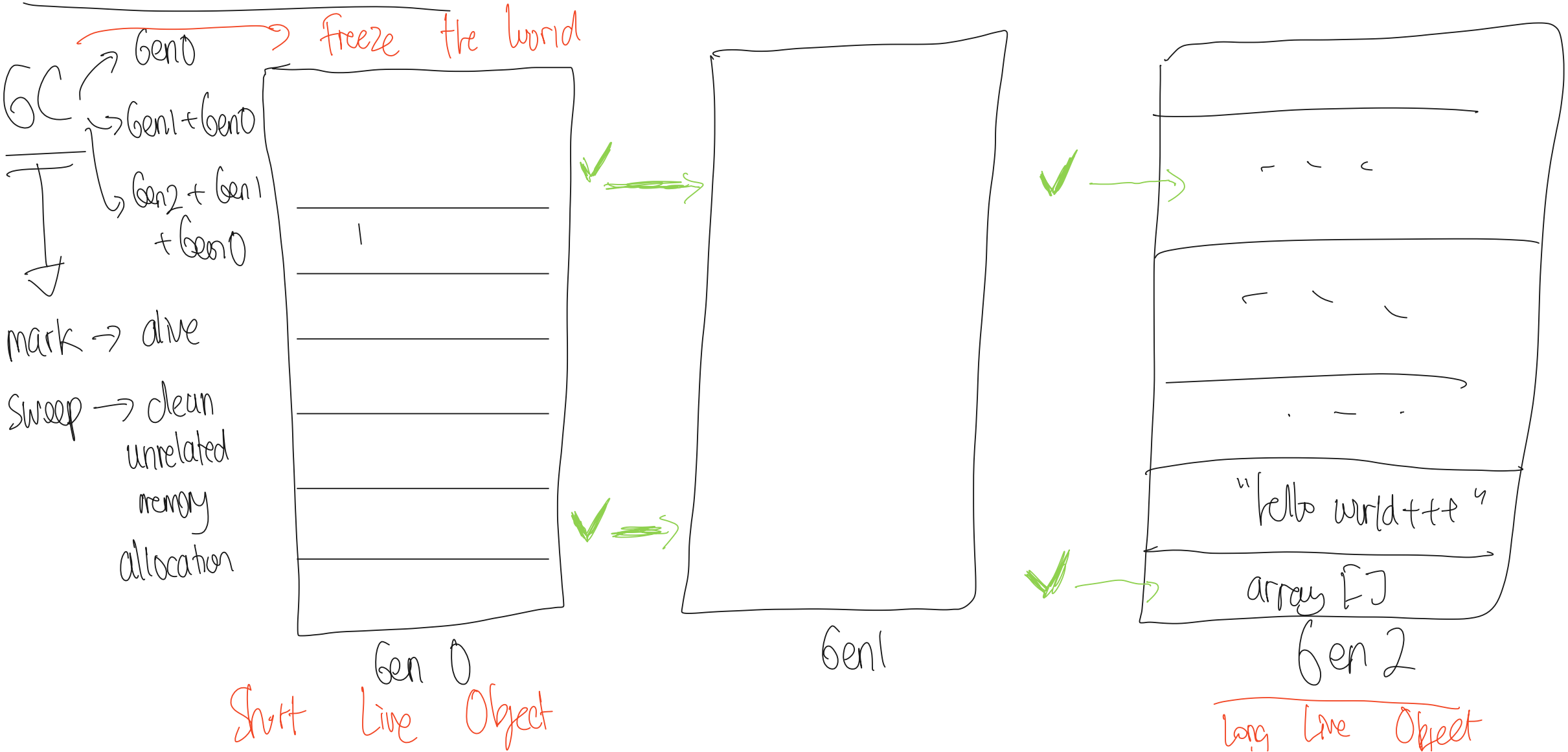
62+61+60



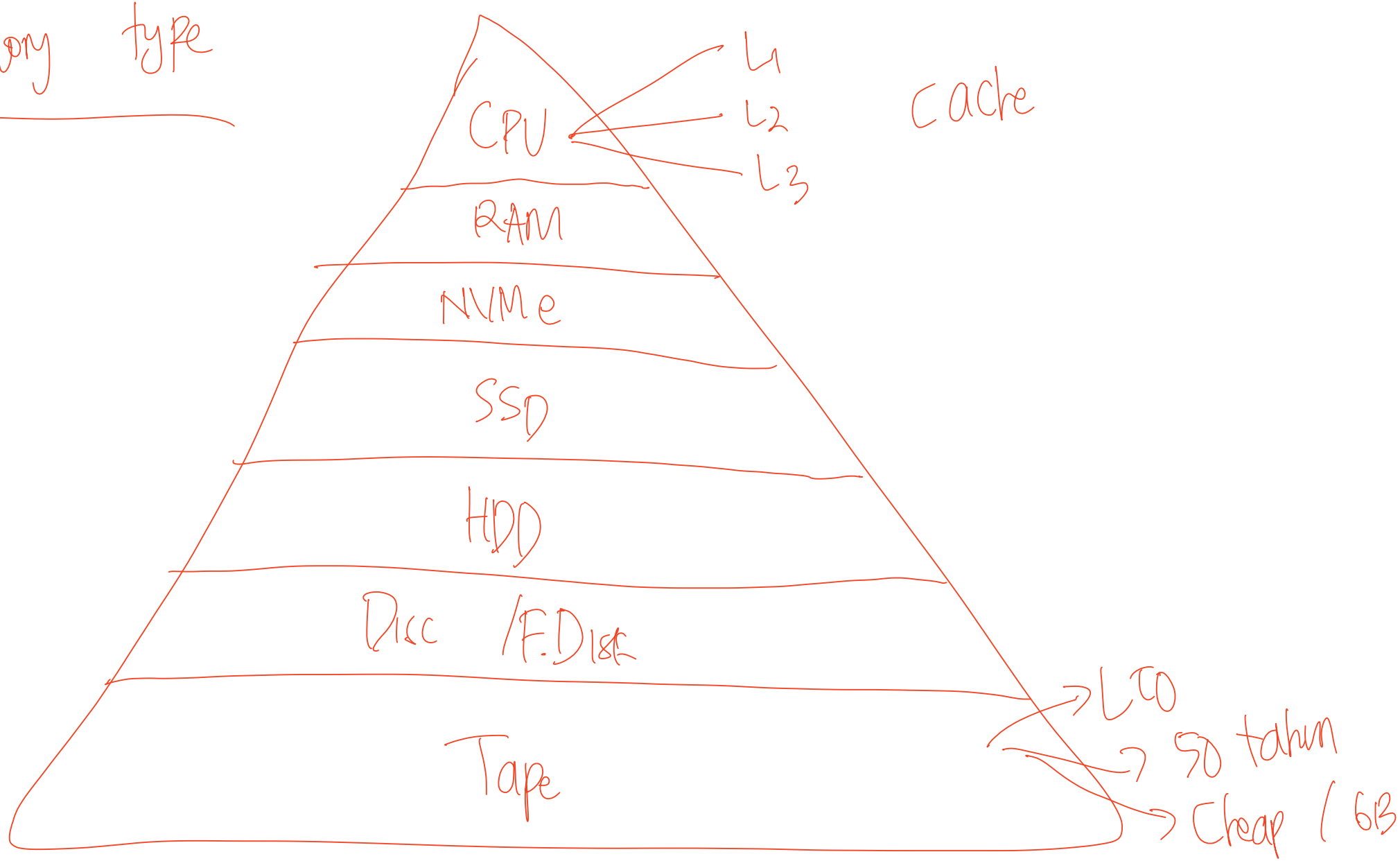
Clean?

- mark = alive
- sweep = Gen

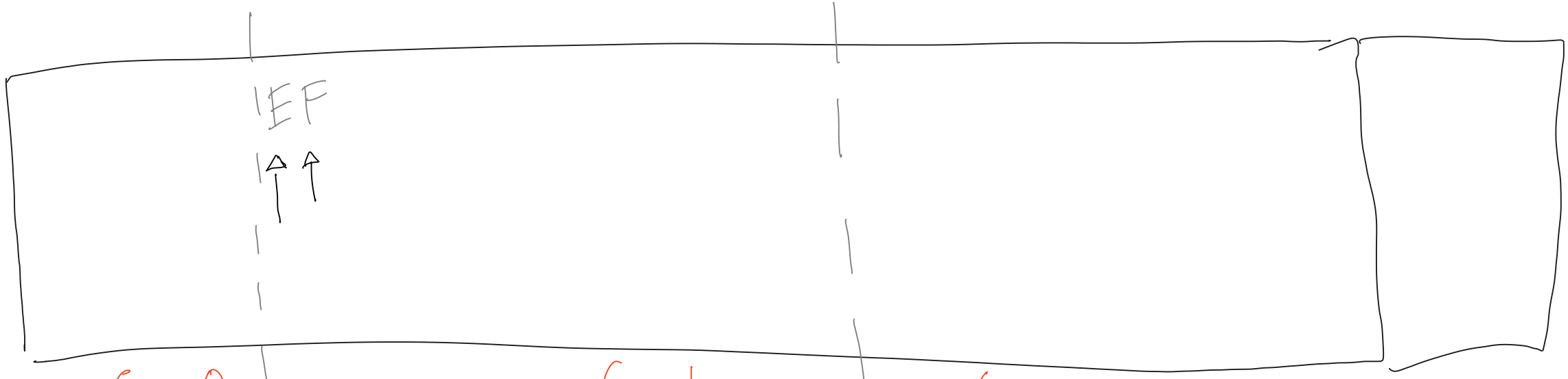
Managed heap



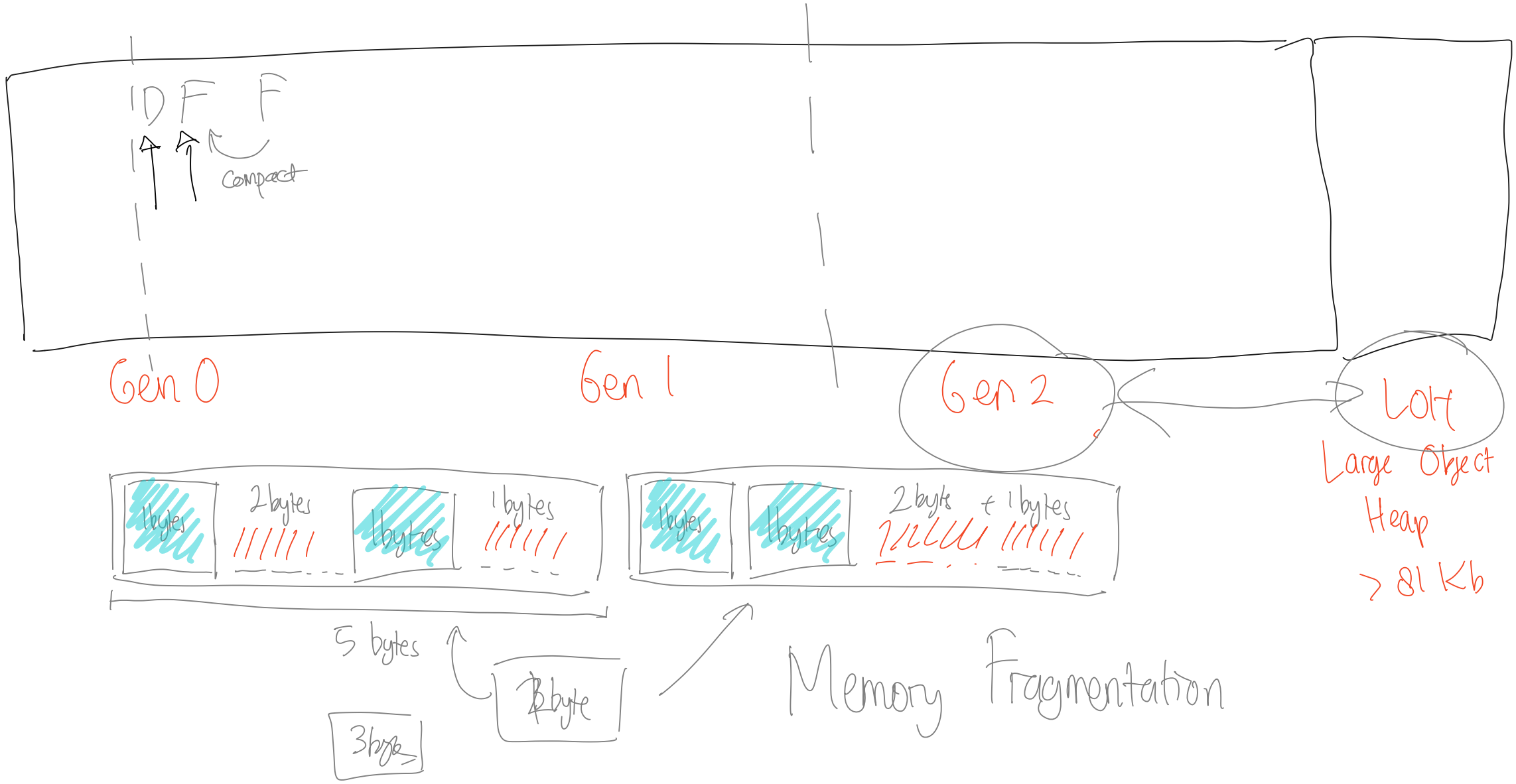
Memory type


CPU
L1
L2
L3
cache
RAM
NVMe
SSD
HDD
Disc / F.Disk
Tape
LTO
SD tahun
Cheap / GB

Garbage Collector → mark
→ sweep
→ Compact



Managed Heap


D F F
Compact
Gen 0
Gen 1
Gen 2
LOH
Large Object Heap
> 81 Kb
1bytes
2bytes
1bytes
1bytes
1bytes
2byte + 1bytes
5 bytes
3byte
Memory Fragmentation

- Finalizers] + Kekurangan (cons)
- Destructor
- IDisposable / Dispose (cons)
- using