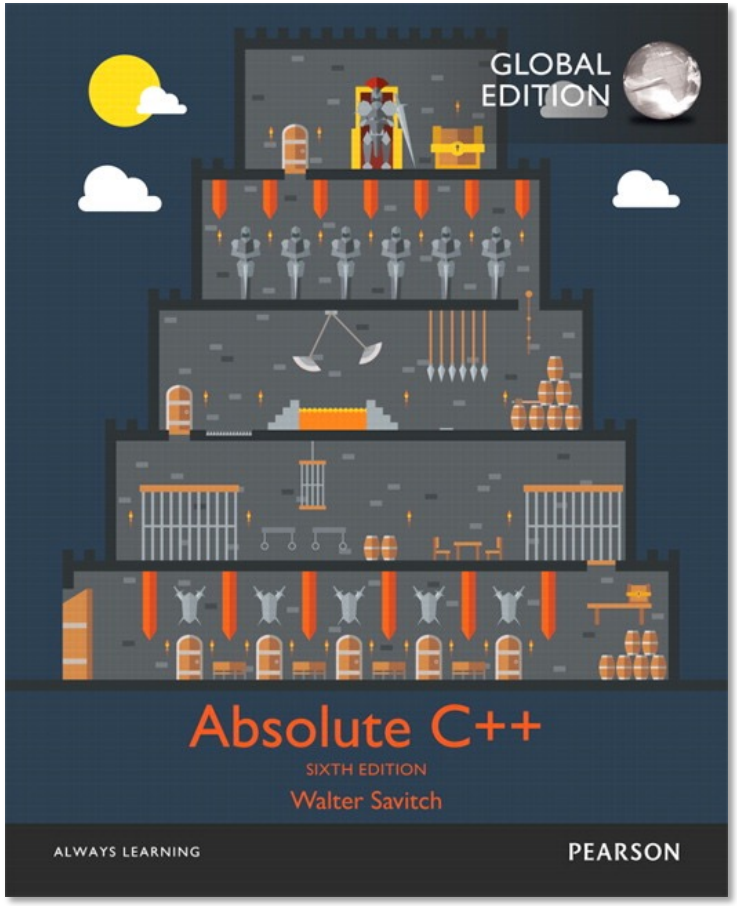
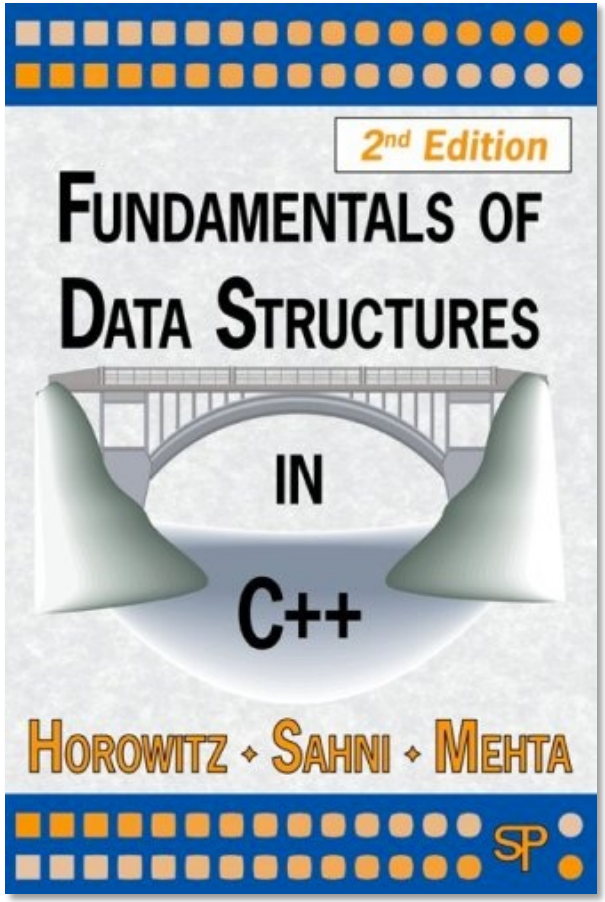
資料結構與物件導向程式設計
OOP & DS

2024 Spring 詹力韋

物件導向(OOP) → 期中考 → 資料結構 (DS) → 期末考



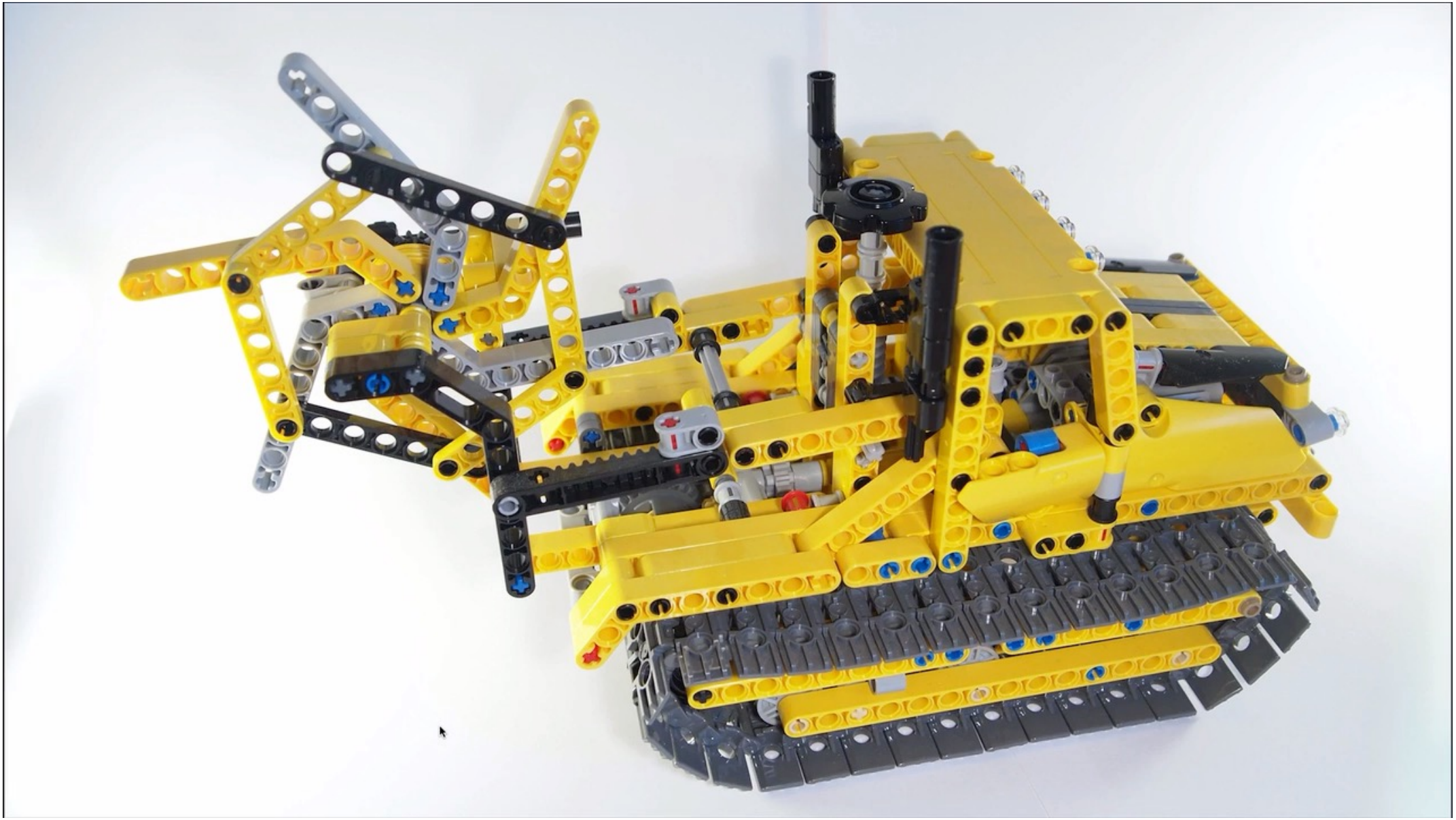


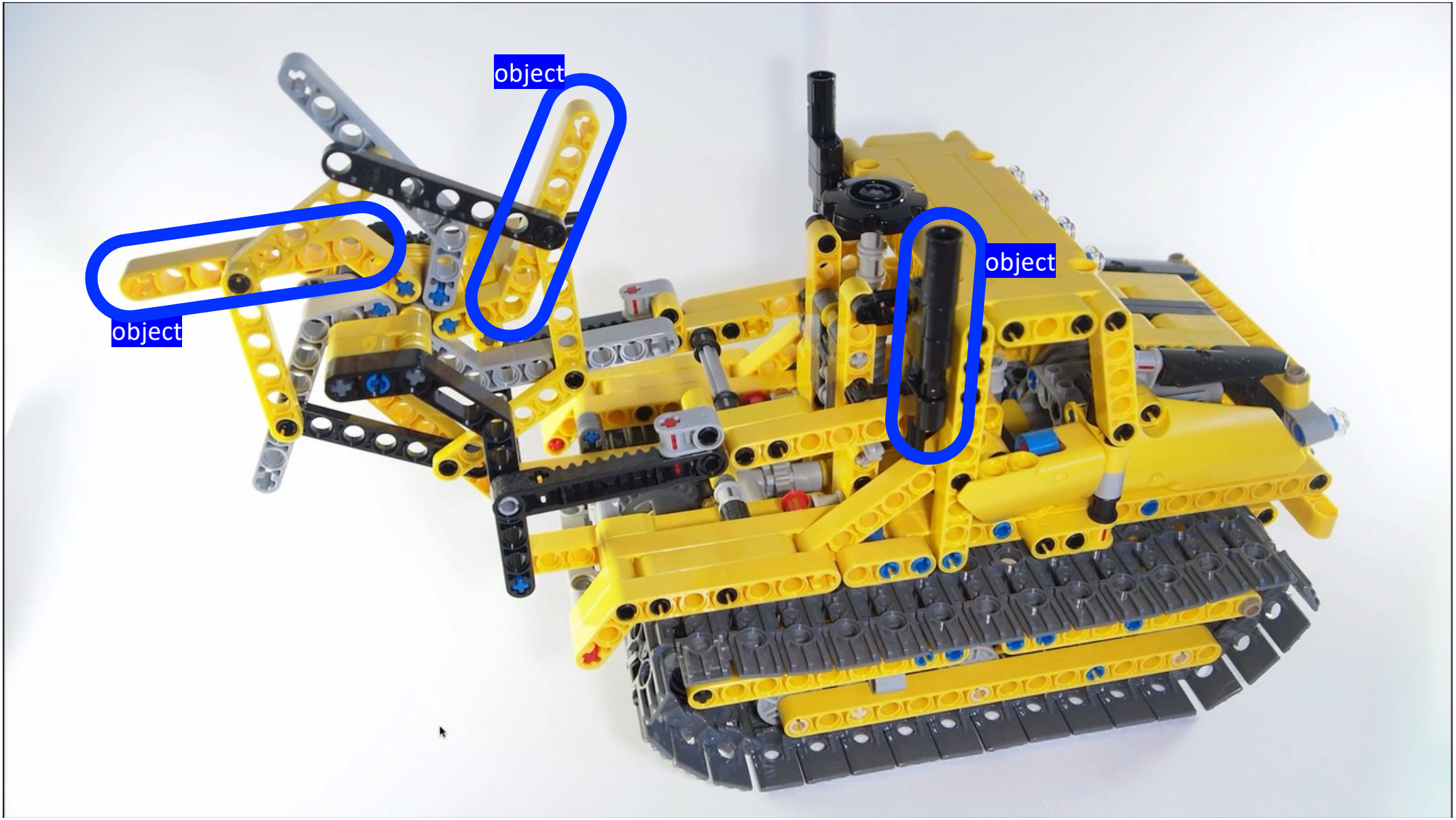
物件導向語言 (Objected-Oriented)

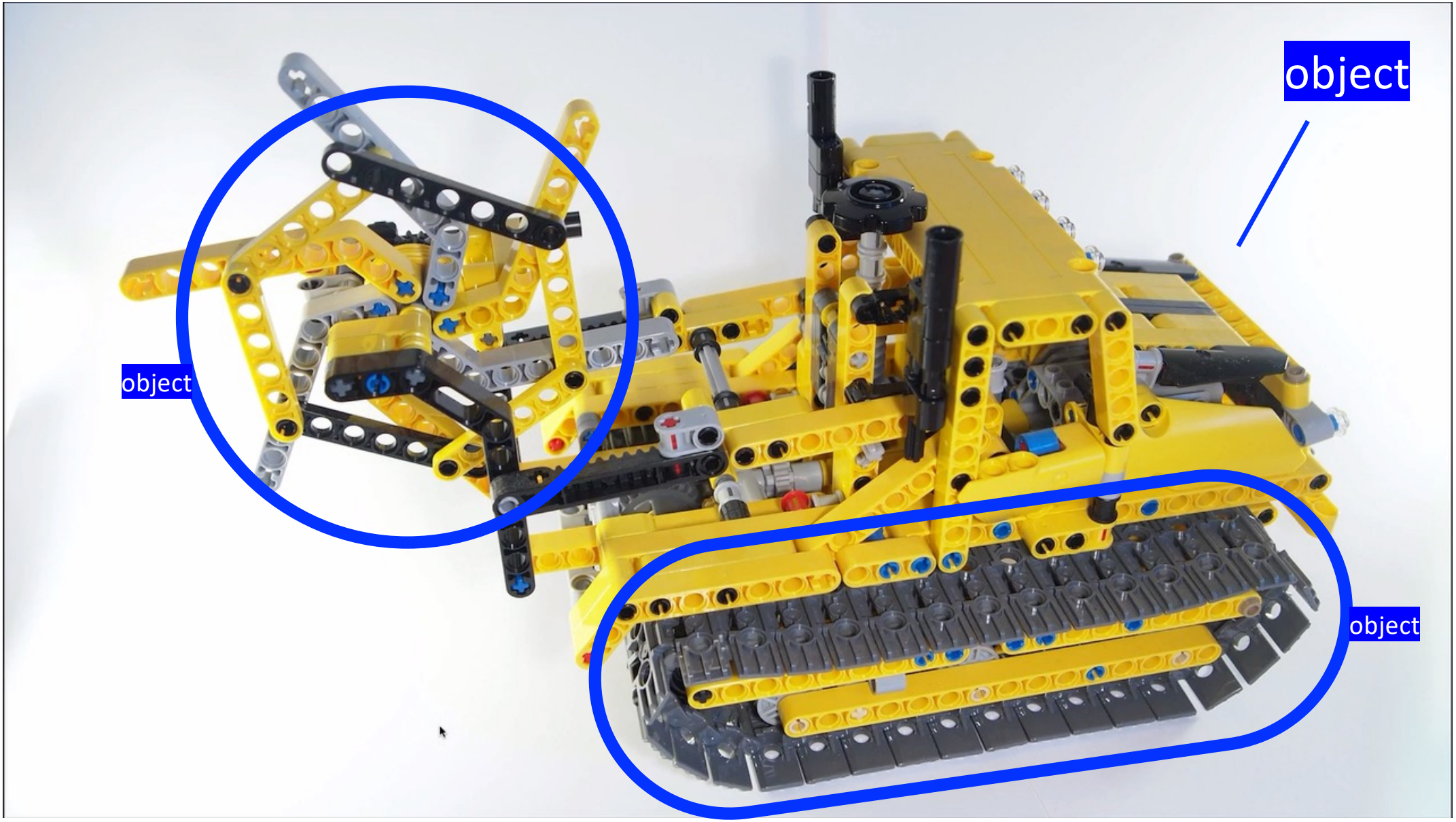
OOP: is a type of computer programming (software design) in which programmers define the data type of a data structure (object), and also the types of operations (functions) that can be applied to the data structure.

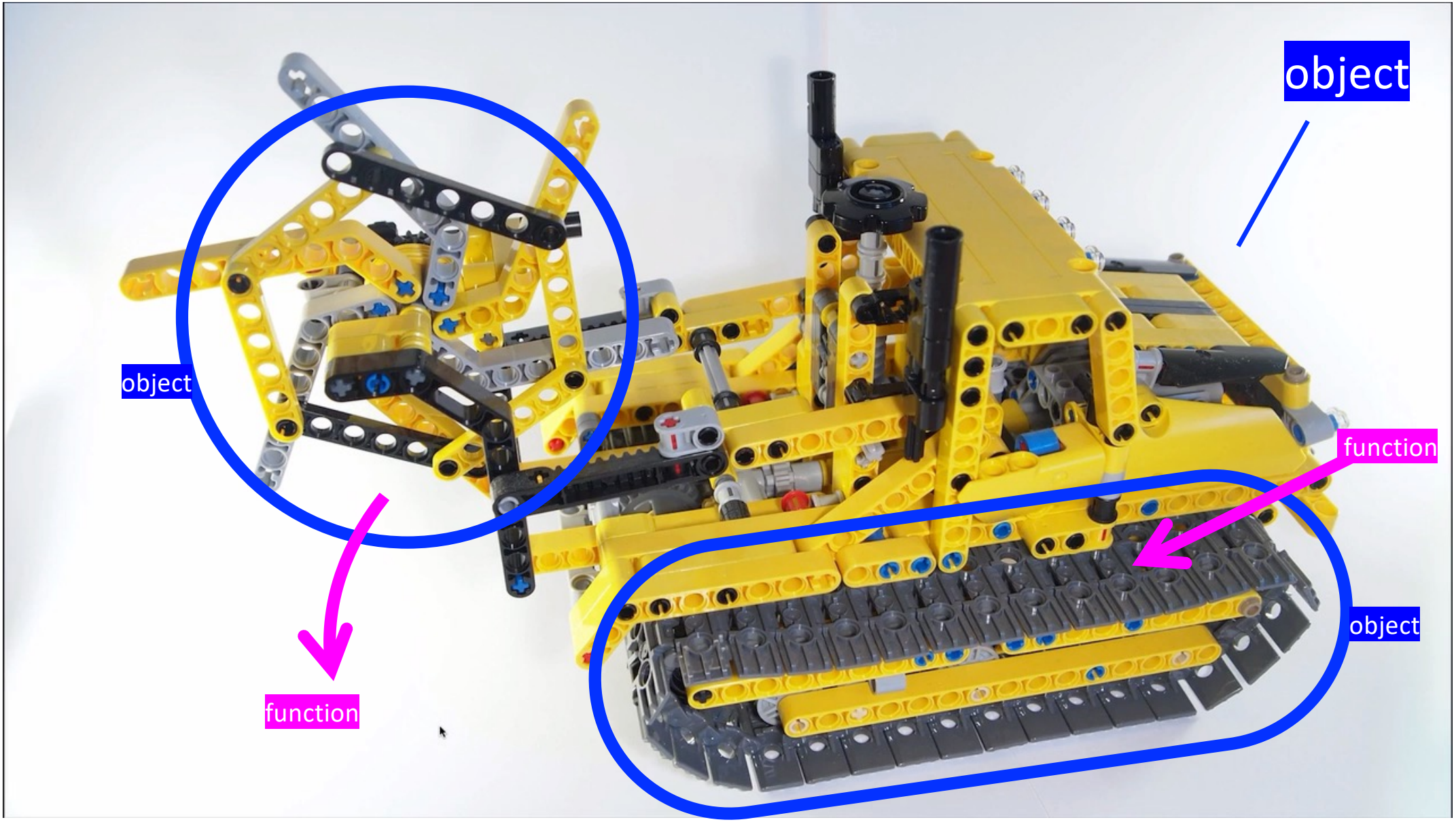
資料結構 (Data Structure)

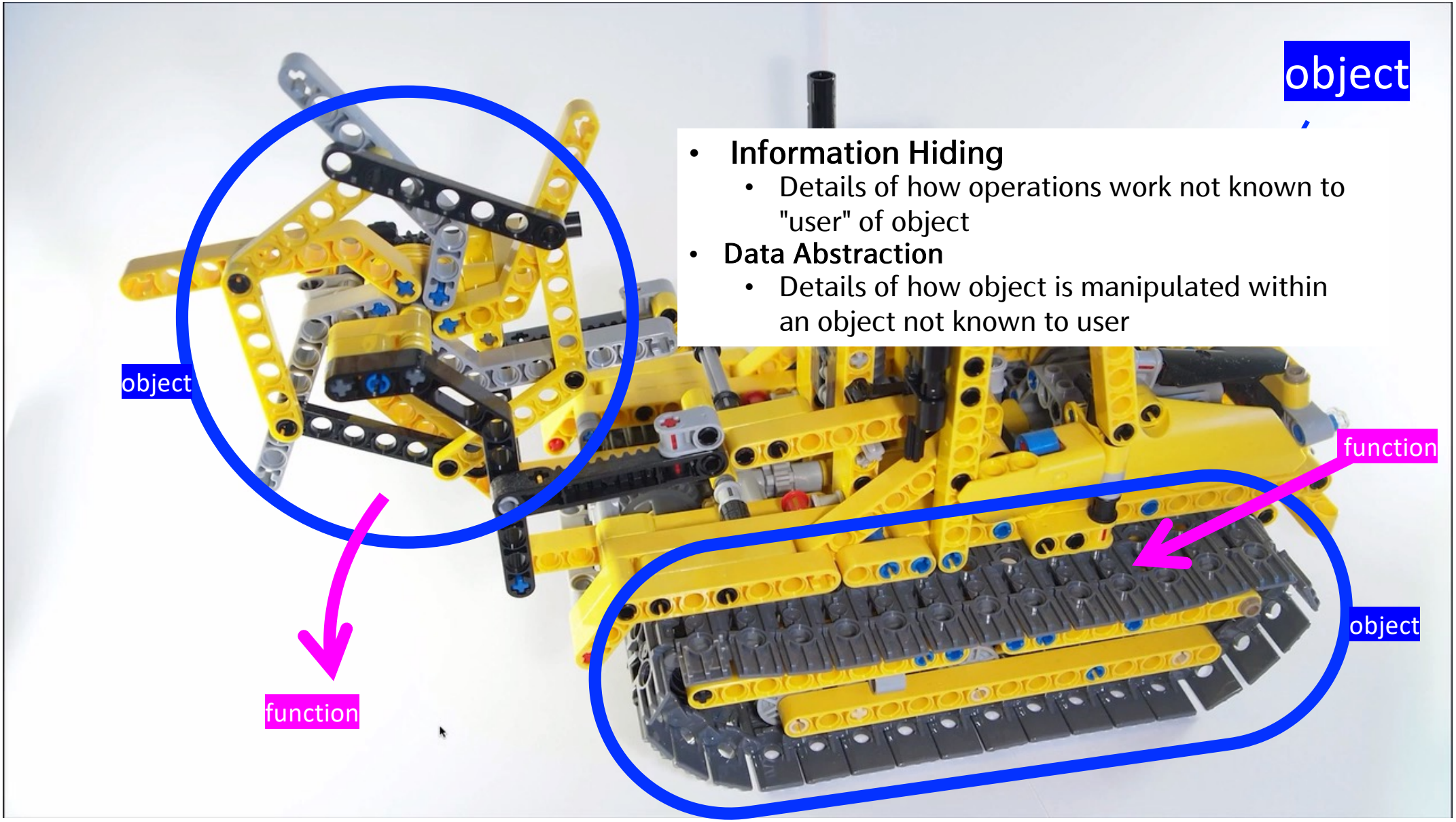
Data Structure: is a way to organize data in a way that enables it to be processed in an efficient time.




object
object
object


object
object
object


object
object
function
function
object


object
Information Hiding
Details of how operations work not known to "user" of object
Data Abstraction
Details of how object is manipulated within an object not known to user
object
function
function
object

資料結構 (Data Structure)

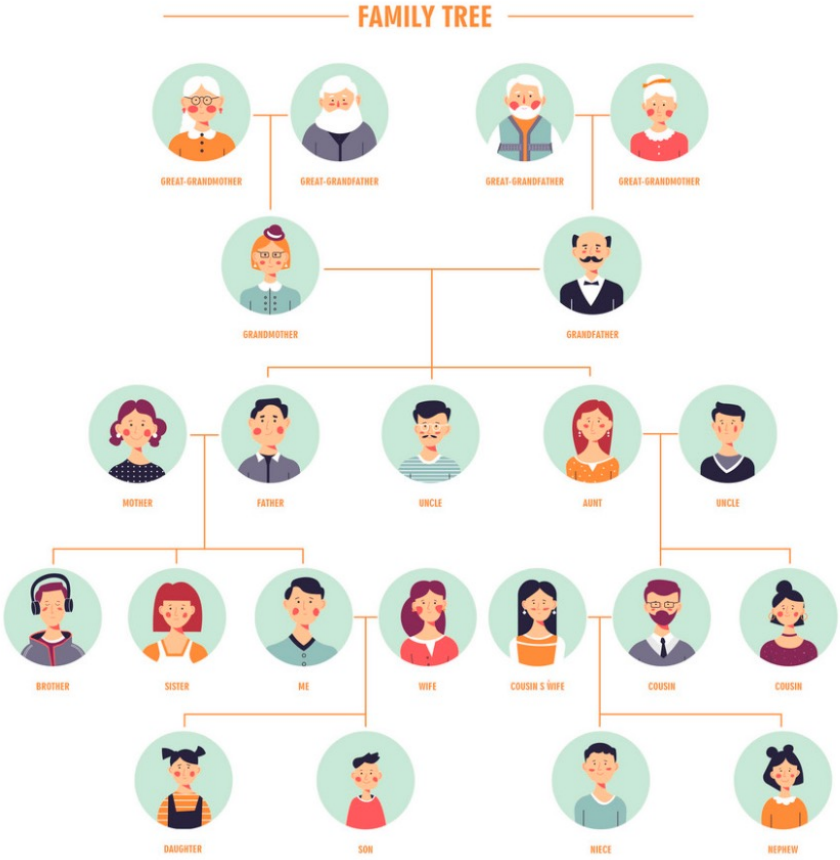
- Data Structure: to organize data in a way that enables it to be processed in an efficient time.
- Common Data Structure are:
  - Array
  - Linked List
  - Stack
  - Queue
  - Tree
  - Hashing
  - Graph, etc.

Data Structures in Everyday Life

- Queue: people lining up for elevator

Data Structures in Everyday Life

- Tree: parent-child relationship

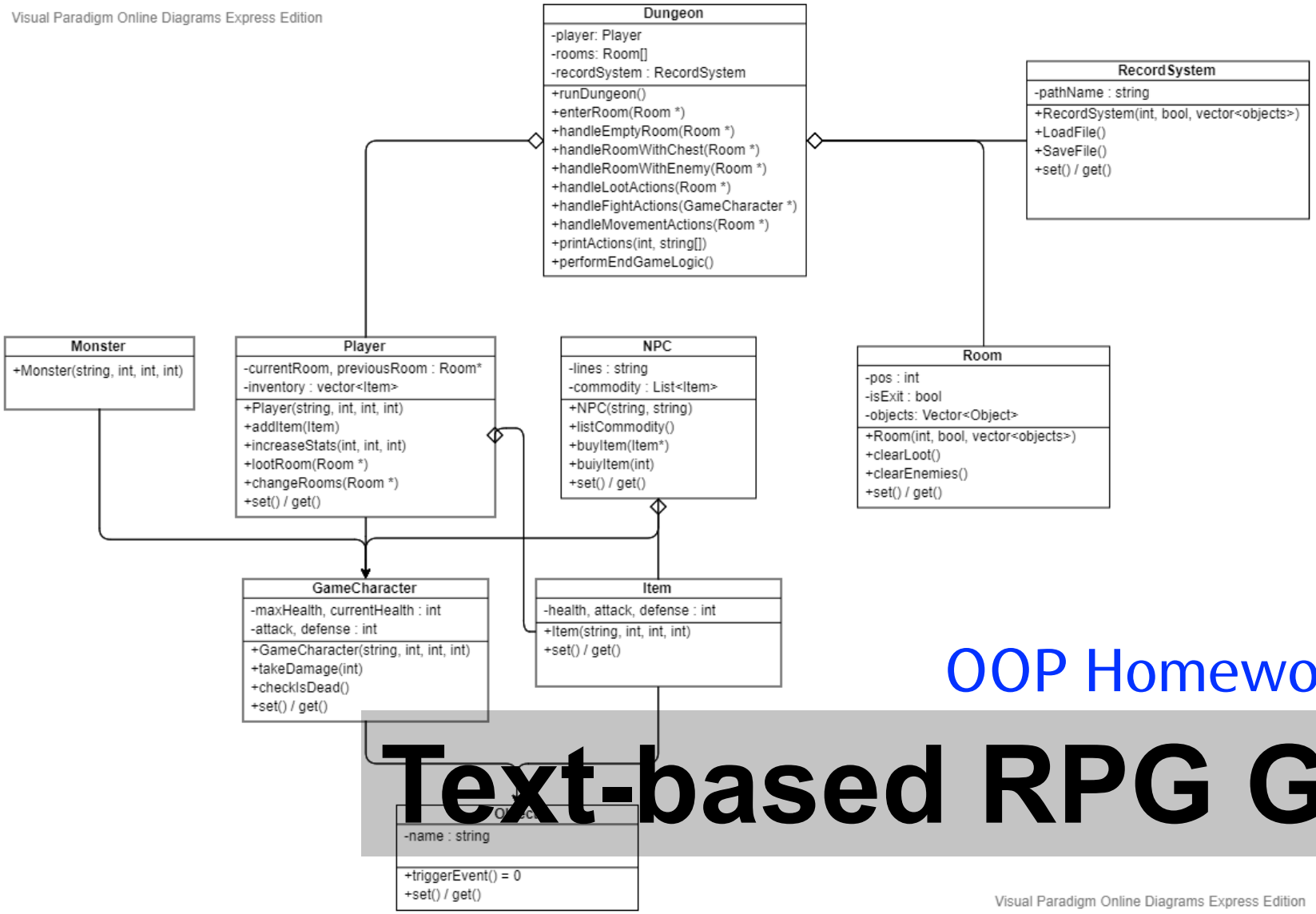


Data Structures in Everyday Life

- Graph: shortest route from one station to another

Grading

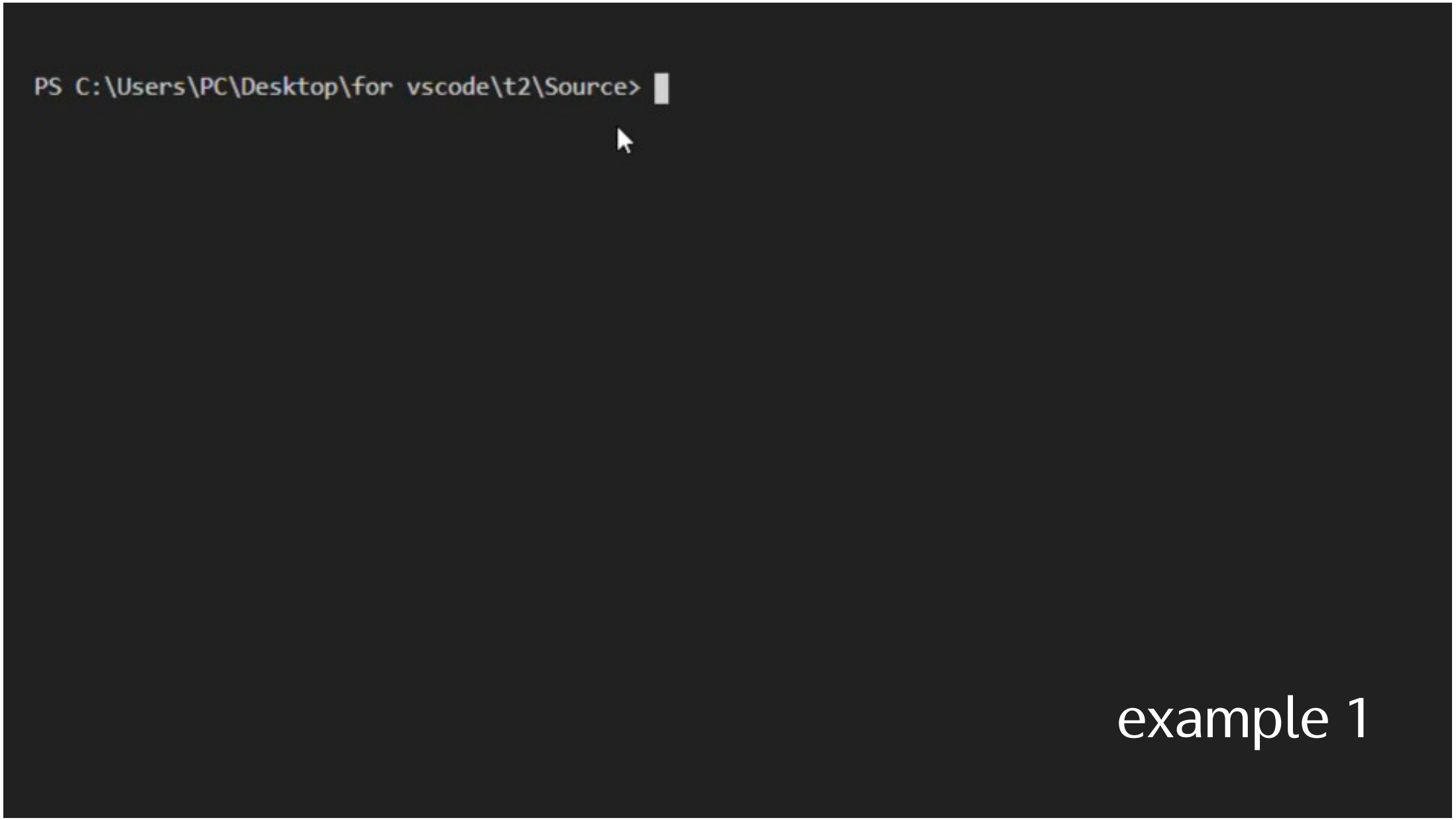
- Regular Lab (20%)
  - demo required; lowest two labs will be dropped
- Homework (OOP: 15%; DS: 10%)
- Midterm Exam (Paper: 20%)
- Final Exam (Paper: 35%)

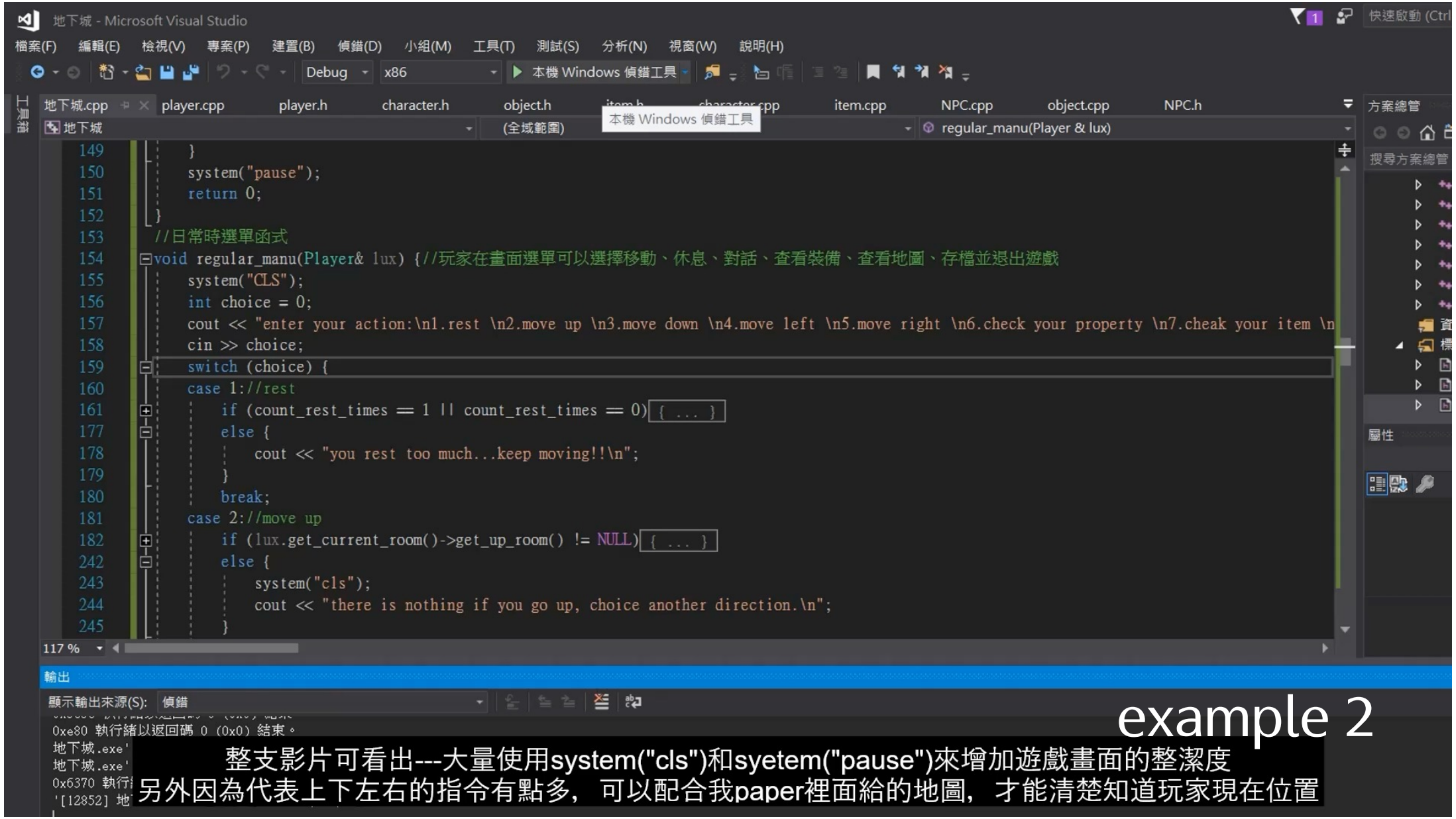


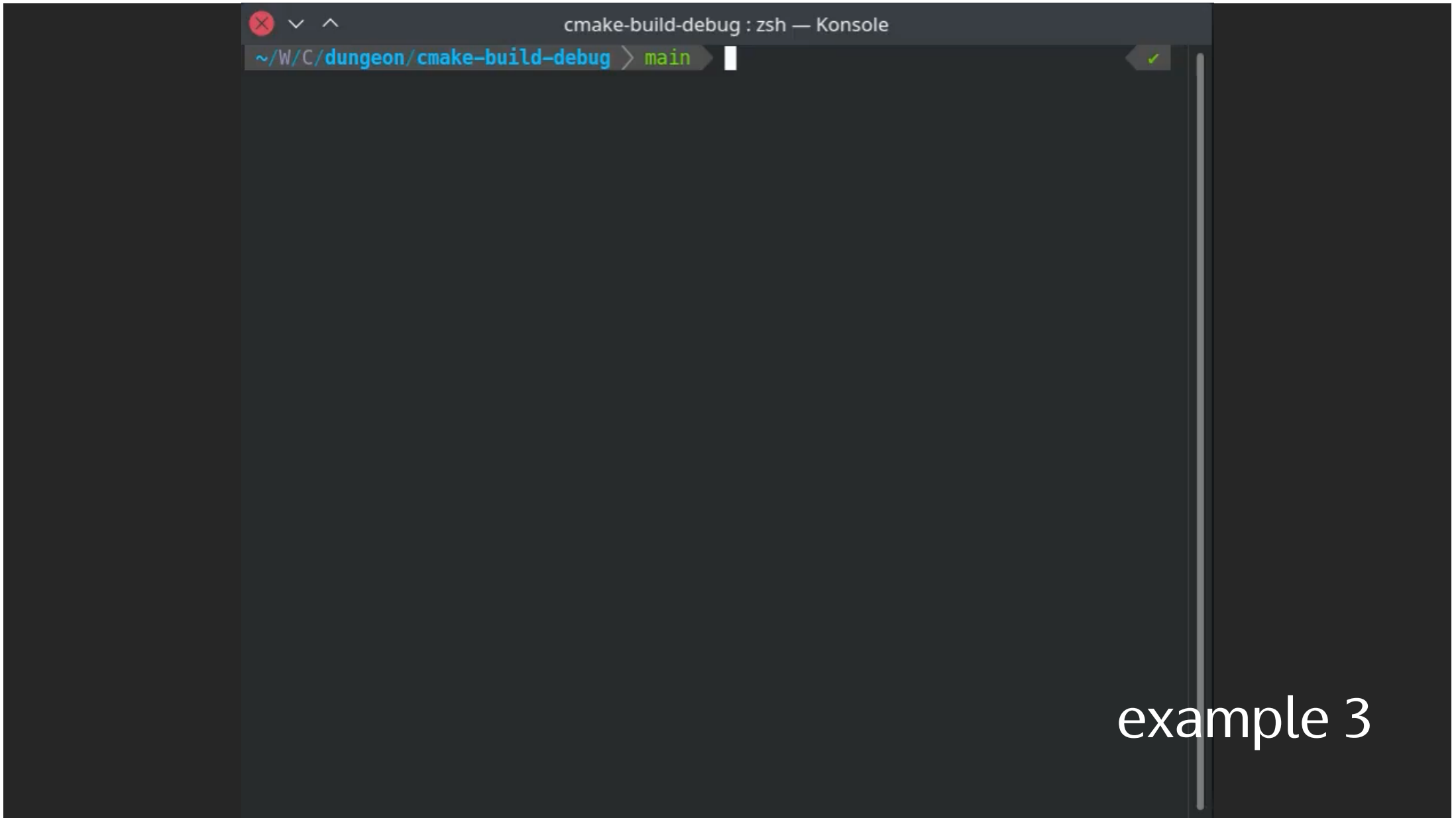
OOP Homework (15%)

Text-based RPG Game

examples from
previous studients


PS C:\Users\PC\Desktop\for vscode\t2\Source>
example 1


地下城 - Microsoft Visual Studio
檔案(F) 編輯(E) 檢視(V) 專案(P) 建置(B) 偵錯(D) 小組(M) 工具(T) 測試(S) 分析(N) 視窗(W) 說明(H)
Debug x86 本機 Windows 偵錯工具
地下城.cpp player.cpp player.h character.h object.h item.cpp NPC.cpp object.cpp NPC.h
地下城 (全域範圍) regular_manu(Player & lux)
149 }
150 system("pause");
151 return 0;
152 }
153 //日常時選單函式
154 void regular_manu(Player& lux) {//玩家在畫面選單可以選擇移動、休息、對話、查看裝備、查看地圖、存檔並退出遊戲
155 system("CLS");
156 int choice = 0;
157 cout << "enter your action:\n1.rest \n2.move up \n3.move down \n4.move left \n5.move right \n6.check your property \n7.cheak your item \n
158 cin >> choice;
159 switch (choice) {
160 case 1://rest
161 if (count_rest_times == 1 || count_rest_times == 0){ ... }
177 else {
178 cout << "you rest too much...keep moving!!\n";
179 }
180 break;
181 case 2://move up
182 if (lux.get_current_room()->get_up_room() != NULL){ ... }
242 else {
243 system("cls");
244 cout << "there is nothing if you go up, choice another direction.\n";
245 }
117 %
輸出
顯示輸出來源(S): 偵錯
0xe80 執行緒以返回碼 0 (0x0) 結束。
地下城.exe'
地下城.exe'
0x6370 執行
'[12852] 地
example 2
整支影片可看出---大量使用system("cls")和syetem("pause")來增加遊戲畫面的整潔度
另外因為代表上下左右的指令有點多， 可以配合我paper裡面給的地圖，才能清楚知道玩家現在位置


cmake-build-debug : zsh — Konsole
~/W/C/dungeon/cmake-build-debug main
example 3

Lab Schedule for OOP

- C++

1.no schedule
2.Lab rules; simple io
3.class; operator overloading
4.inheritance
5.linked list
6.polymorphism
7.string; file; exception handling
8.QA
9.no schedule

- Text-based PRG Game

1.no schedule
2.UML
3.game char. (player; NPC; monster)
4.object; room; item
5.dungeon
6.dungeon
7.record system; bonus idea
8.QA
9.video demo

這週沒有上機