프로그래밍 나두 할 수 있다!
I CAN DO PROGRAMMING

•••

세번째 모임
~Third Meeting~

오늘의 주제

- 문제 같이 풀어봅시다!
  - Private vs. public
  - Deep copy
  - Shallow copy
- C code 파일 문제도 같이 풀어봅시다!
  - Data type transformation
  - Iteration: do-while, while, for loop
- RAM
- Array
  - Scores.c
  - Brick.c
  - hi.c

```c
#include <stdio.h>
#include <cs50.h>

int main(){
    int a = 5;
    int b = 3;
    printf("%d\n", a / b);
    printf("%f\n", (float) a / b);
    printf("%f\n", a * 1.0 / b);

}

```

Q1. printf("%d\n", a / b); printf("%f\n", (float) a / b); printf("%f\n", a * 1.0 / b); 코드를 실행시켜보고 차이점이 무엇인지 설명해주세요.

Q2. 세개의 코드가 차이점을 갖는 이유가 무엇일까요?

Q3. printf("%f\n", (float) a / b); 여기서 float가 하는 역할이 무엇인가요?

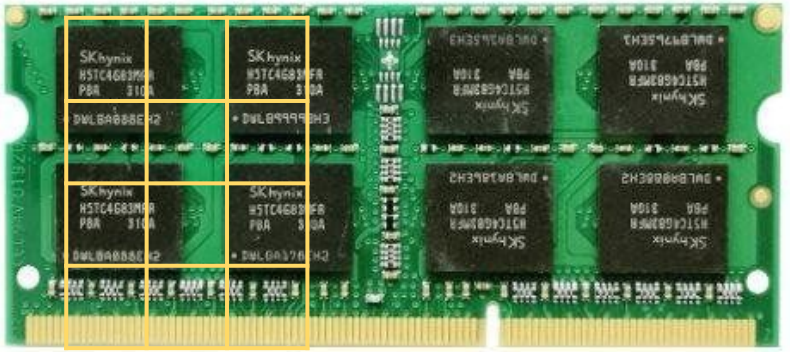
```c
//Abstraction and scope

#include <cs50.h>
#include <stdio.h>

int get_positive_integer(void);

int main(void)
{
    int i = get_positive_integer;
    printf("%i\n", i);
}

//prompt user for positive integer
int get_positive_int(void)
{
    int n;
    do
    {
        n = get_int("Positive Integer: ");
    //int n = get_int Does it work?
    }
    while (n < 1);
    return n;
}

```

Q1 do-while 대신 while과 for을 이용해서 같은 결과를 만들어보세요.

Q2 // 는 어떤 역할인가요?

RAM

Random access memory

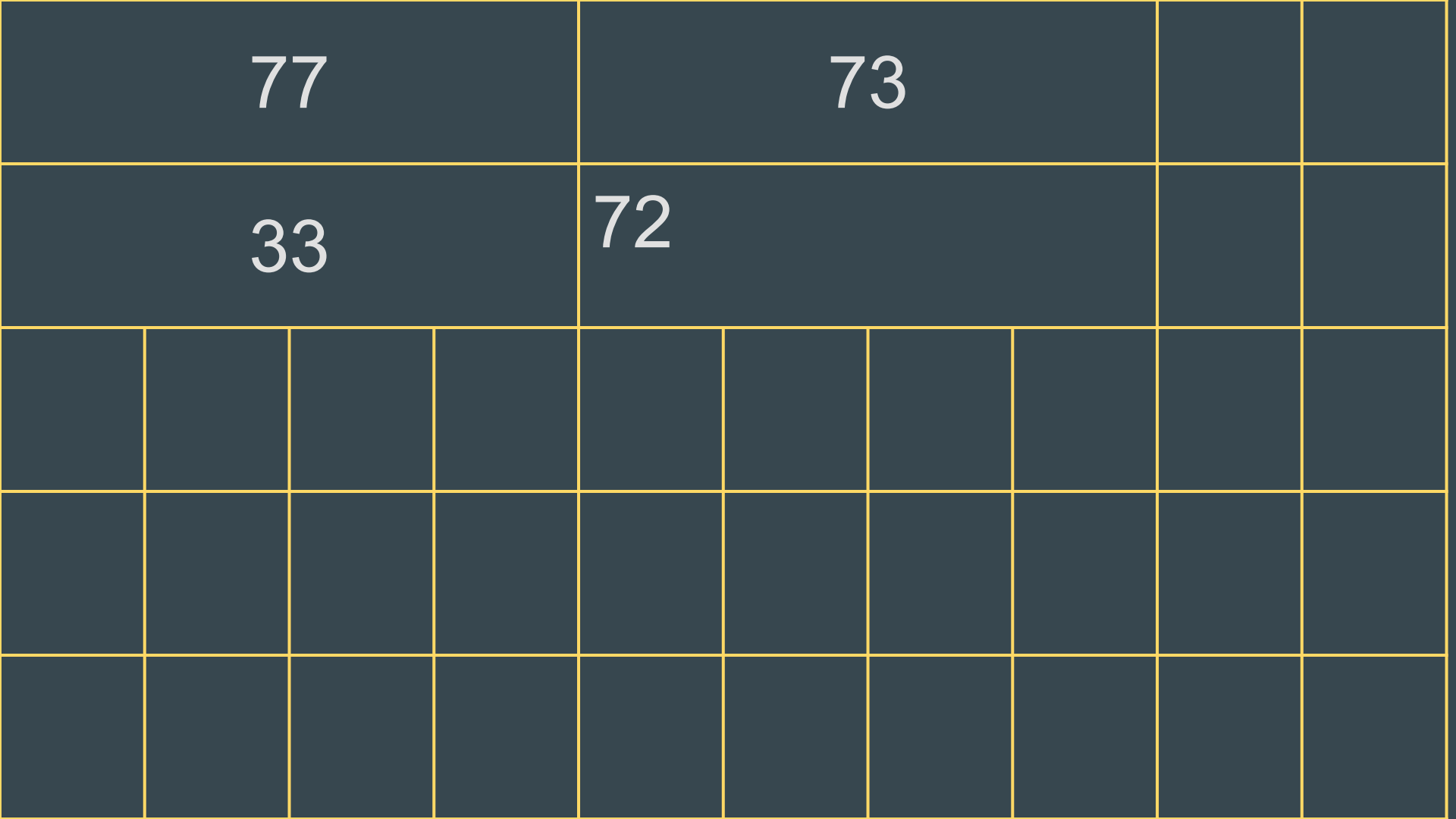


```
int score1 = 72;

int score2 = 73;

int score3 = 33;
```

72

72

73


77
73
33
72

Array

72
score[0]

73
score[1]

33
score[2]

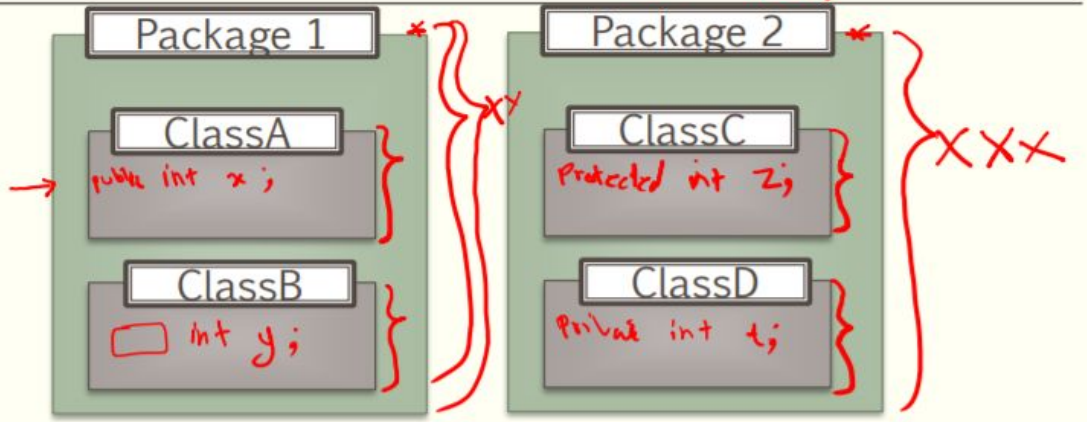
어떤 보안 구멍이 있을까요?

```java
public class SelfDrivingCar {
    private String make;
    private String model;
    private int color;
    private char[] plateNumber;
    private int maxAllowedSpeed;

    public String getMake() {
        return make;
    }
    public void setMake(String make) {
        this.make = make;
    }

    public char [] getPlateNumber() {
        return plateNumber;
    }
    public void setPlateNumber(char[] plateNumber) {
        this.plateNumber = plateNumber;
    }
}
```

```java
public class UseCar {
    public static void main(String [] args) {
        SelfDrivingCar myCar = new SelfDrivingCar();
        myCar.setMake("Toyota");
        char [] pno = {'x','x','x','x','1','1','1'};
        myCar.setPlateNumber(pno);
        pno[0]='s';
        System.out.println(myCar.getPlateNumber());
    }
}
```

Access Modifiers

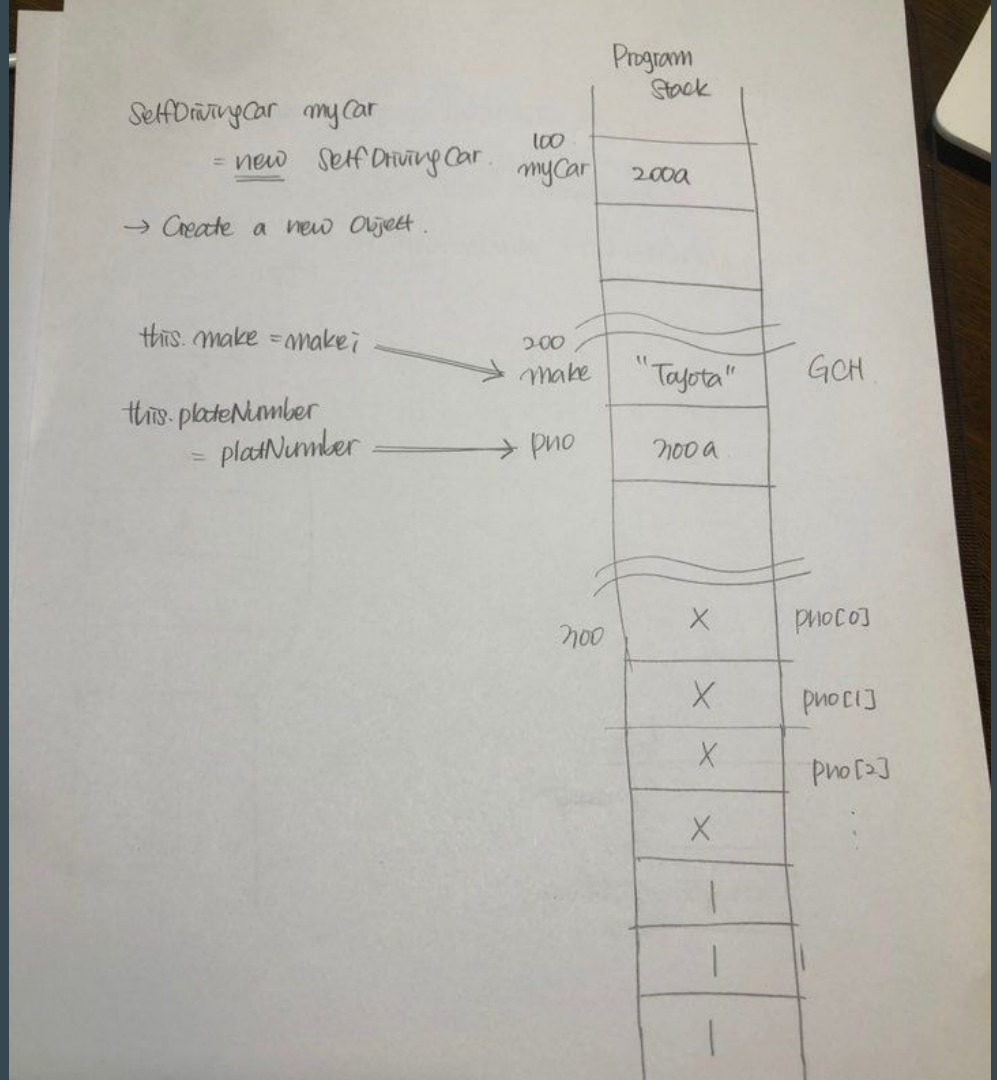
- To define an access level for a variable, method or constructors `access modifiers` are used.
- Access modifiers visibility:



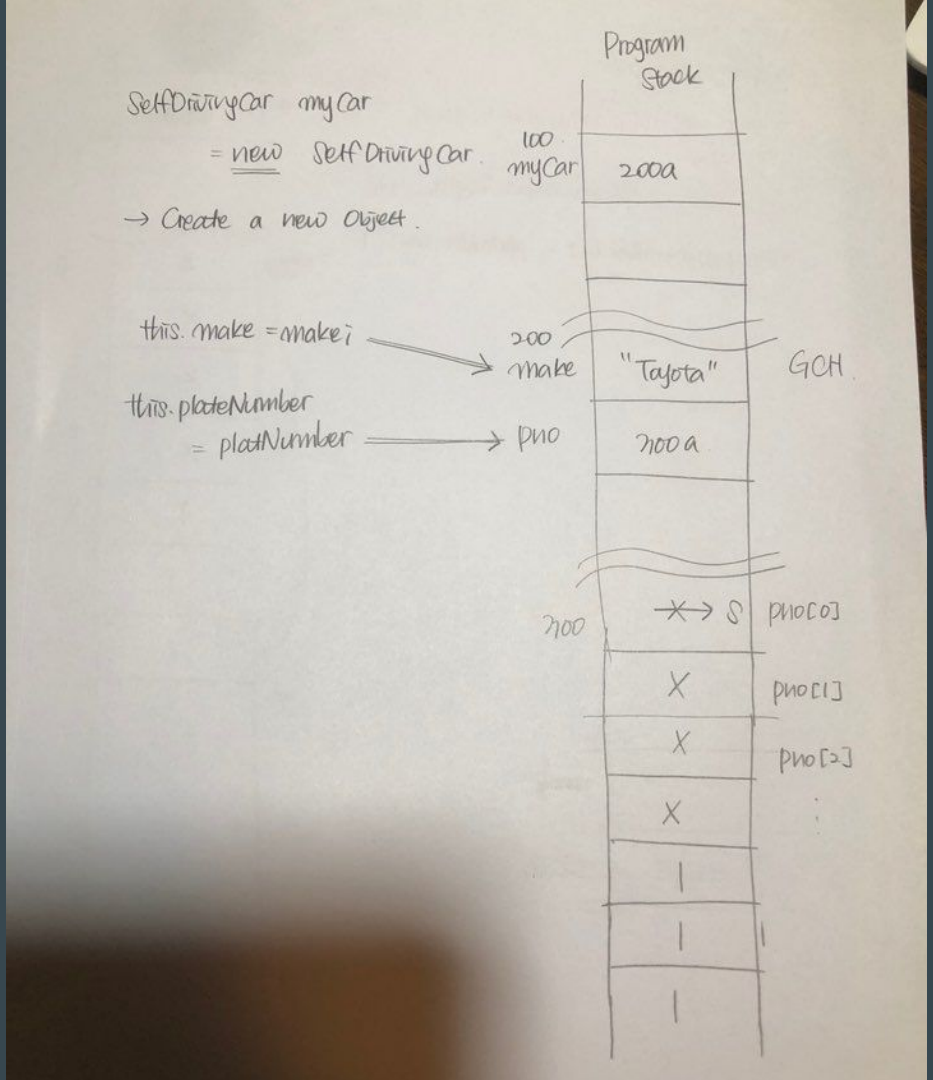
Access Modifier	class	subclass	package	World (outside the package)
public	Y	Y	Y	Y
No access modifier	Y	Y	Y	N
protected	Y	Y	N	N
private	Y	N	N	N

This column will be revisited later

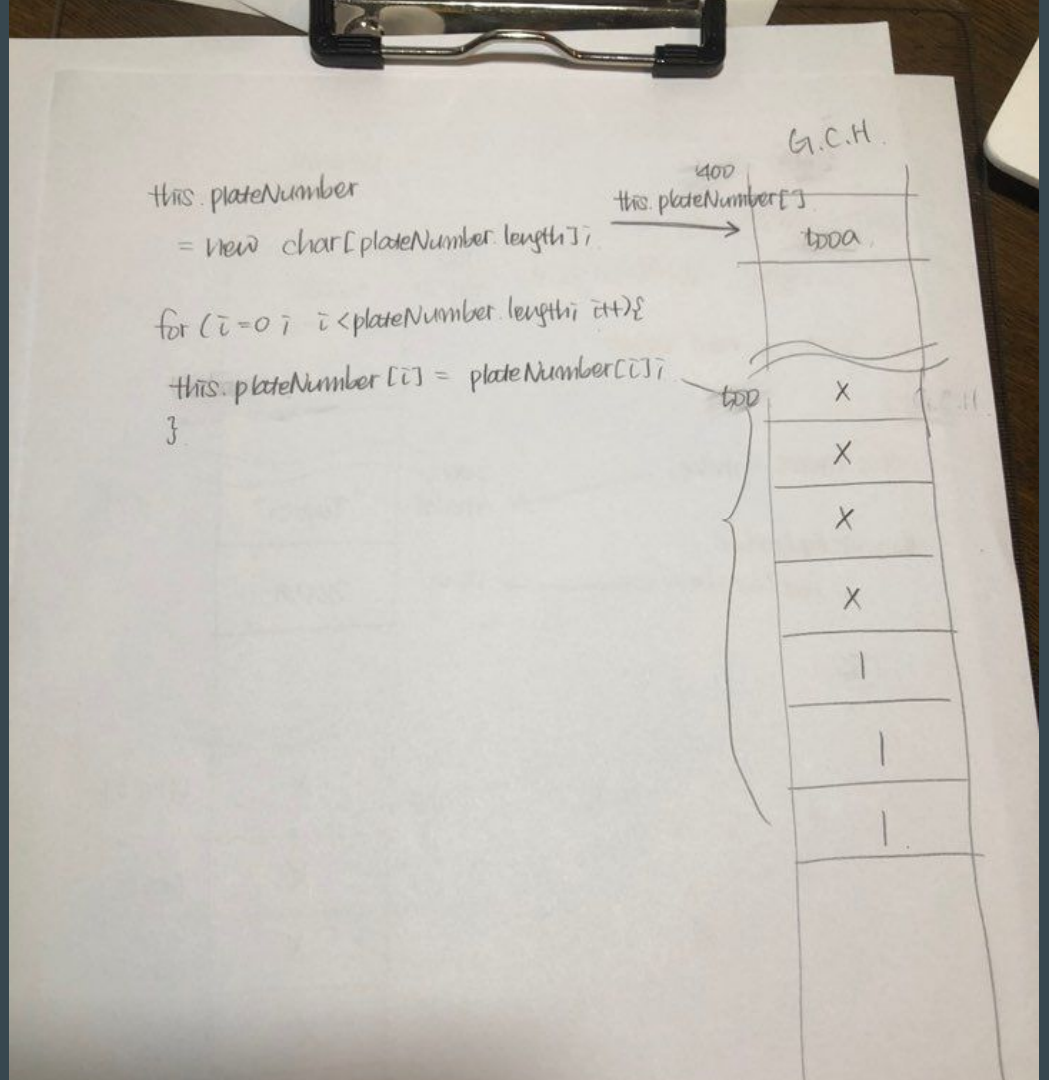
Shallow copy



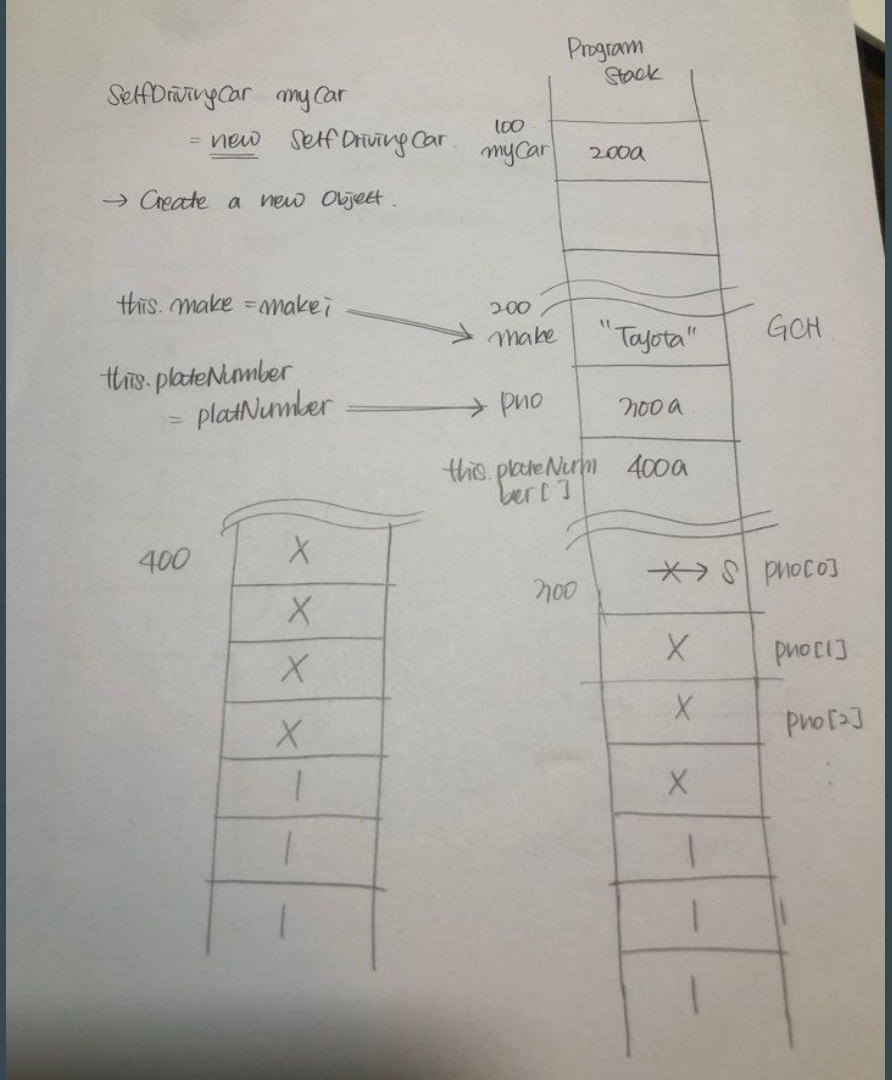
Shallow copy



Deep copy


this.plateNumber
= new char[plateNumber.length];
for(i=0; i<plateNumber.length; i++){
this.plateNumber[i] = plateNumber[i];
}
G.C.H.
400
this.plateNumber[]
1000
500
X
X
X
X
1
1
1

Deep copy


SelfDrivingCar myCar
= new SelfDrivingCar.
→ Create a new object.
this.make = make;
this.plateNumber
= plateNumber
Program
Stack
100
myCar
2000
200
make
"Tayota"
GCH
pno
7000
this.plateNumber[]
4000
400
7100
pno[0]
pno[1]
pno[2]