

Stack Memory

```
main() {
```

```
    int age;
```

```
    age = 12;
```

```
    doWork();
```

```
}
```



```
doWork() {
```

```
    float weight = 80;
```

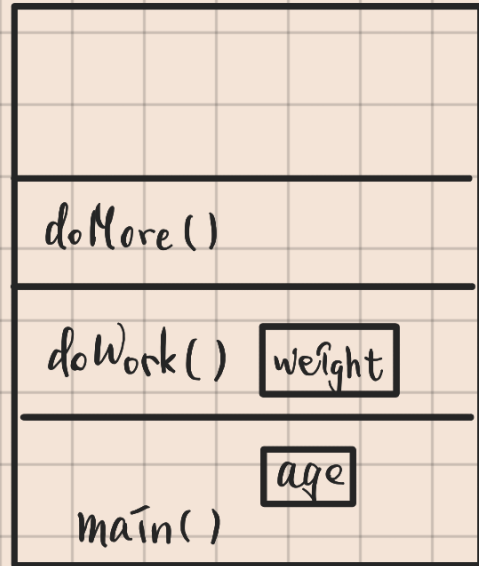
```
    doMore();
```

```
}
```

release
once
func
finish



Stack



↑ memory
piling
up

Heap Memory

```
main() {
```

```
    Car myCar;
```

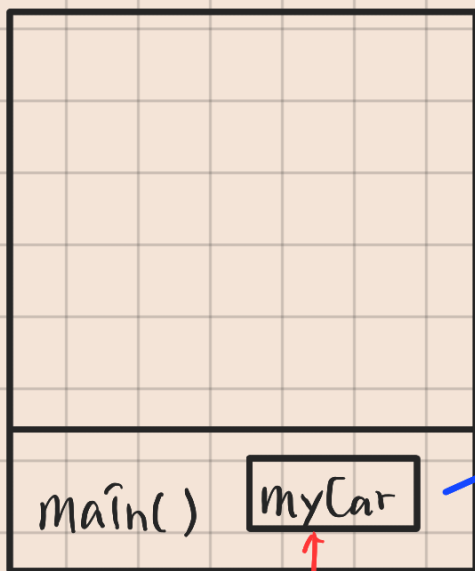
```
myCar = new Car();①
```

reference variable
(store address)

```
myCar = new Car();②
```

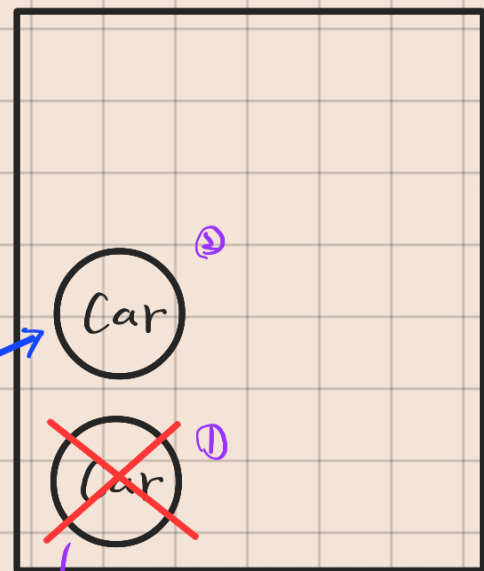
```
}
```

Stack



storing address

Heap



After myCar points to a new car obj, ① will be removed

Garbage collection in Heap: looking for objects that no one is pointing to, and remove them.

```
main() {
```

```
    Car myCar = new Car();
```

```
    myCar.hp = 120;
```

```
    Engine bigEngine = new Engine();
```

```
    myCar.myEngine = bigEngine;
```

```
}
```

```
Class Car {
```

```
    int hp;
```

```
    Engine myEngine;
```

```
}
```

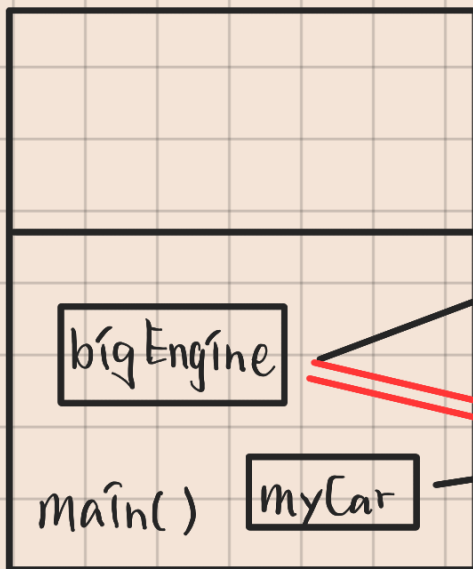
reference
var

在 Heap 創 Engine obj
Heap Engine 位址

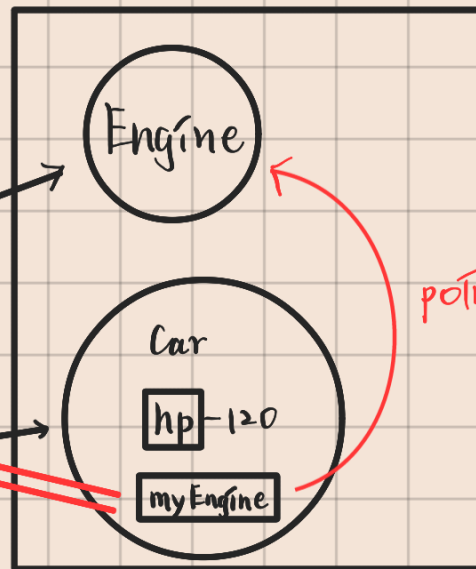
把 Heap Engine 位址給

Car obj 的 reference obj
myEngine

Stack



Heap



assign

pointing

