

Swift class

宣告類別(class)型態Course，第一個字母通常使用大寫，其餘小寫。本題的類別有兩個欄位，一個是整數型態的變數id，另一個是字串型態的變數title。請在初始方法中，設定兩個欄位的初始值

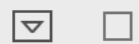
```
1 import UIKit
2 class Course{
3     var id: Int
4     var title: String
5     init(id: Int, title: String){
6                  = id
7                  = title
8     }
9 }
```


請在第五行設定初始方法的兩個輸入參數
並在第十行使用初始方法設定類別欄位初始值

```
1 import UIKit
2 class Mobile{
3     var make: String
4     var model: String
5     init( ) {
6         self.make = make
7         self.model = model
8     }
9 }
10 let myMobile = Mobile.(make: "Taiwan",
11     model: "5s")
11 print(myMobile.make)
```

Mobile

"Taiwan\n"



Taiwan

在第十一行將mobile宣告為類別變數，內容參考myMobile。

```
1 import UIKit
2 class Mobile{
3     var make: String
4     var model: String
5     init( ) {
6         self.make = make
7         self.model = model
8     }
9 }
10 let myMobile = Mobile.(make: "Taiwan",
11     model: "5s")
12 print(myMobile.model)
```

Mobile

Mobile

"5s\n"



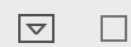
5s

請在第十二行將myMobile變數的model欄位內容改為"8s"。

因為mobile的參考型態為myMobile，所以第13行印出mobile.model為更新後的內容

```
1 import UIKit
2 class Mobile{
3     var make: String
4     var model: String
5     init(make: String, model: String){
6         self.make = make
7         self.model = model
8     }
9 }
10 let myMobile = Mobile.init(make: "Taiwan", model: "5s")
11 var                     
12                      = "8s"
13 print(myMobile.model)
```

Mobile	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
Mobile	<input type="checkbox"/>
"8s\n"	<input type="checkbox"/>



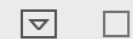
8s

本題Mobile為結構(struct)型態，第十一行將mobile的內容指定為myMobile的內容
雖然第十二行更新myMobile.model，列印mobile.model的結果仍然為5s

```
1 import UIKit
2 struct Mobile{
3     var make: String
4     var model: String
5     init(make: String, model: String){
6         self.make = make
7         self.model = model
8     }
9 }
10 var myMobile = Mobile.init(make: "Taiwan", model: "5s")
11 var XXXXXXXXXX
12 myMobile.model = "8s"
13 print(mobile.model)
```



Mobile
Mobile
Mobile
"5s\n"



5s

請在第八行將類別Fullhouse的型態宣告為Straight
類別(class)Fullhouse繼承類別(class)Straight的欄位變數與初始方法，請在第
十二行，使用繼承類別的初始化方法設定欄位three的初始內容

```
1 import UIKit
2 class Straight{
3     var three: Int
4     init(three: Int){
5         self.three = three
6     }
7 }
8 class Fullhouse: Straight {
9     var pair: Int
10    init(pair: Int, three: Int){
11        self.pair = pair
12        super.init(three: three)
13    }
14 }
```


請在第十五行，根據列印結果設定第二欄位的初始內容

```
1 import UIKit
2 class Straight{
3     var three: Int
4     init(three: Int){
5         self.three = three
6     }
7 }
8 class Fullhouse: ██████████ {
9     var pair: Int
10    init(pair: Int, three: Int){
11        self.pair = pair
12        super. ██████████
13    }
14 }
15 let fullhouse = Fullhouse(pair:2, ██████████)
16 print(fullhouse.three)
```

Fullhouse
"9\n"

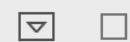


類別(class)Robot有兩個欄位，第一個欄位已經使用default的方式設定，初始化方法只要設定第二個欄位即可。請根據列印結果在第八行設定初始內容

```
1 class Robot{
2     var withArm = true
3     var withLeg: Bool
4     init(withLeg: Bool){
5         [redacted]
6     }
7 }
8 let robot = Robot([redacted])
9 print(robot.withArm, robot.withLeg)|
```



Robot
"true tru...



true true

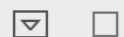
請第三行將欄位withLeg的型態宣告為option布林變數。

本題有兩個初始化方法，第一個初始化方法，僅初始化其中一個欄位，第二個初始化方法，則初始化兩個欄位。

請在第十五行的列印指令中將變數robot2的第二個欄位展開。

```
1 class Robot{
2     var withArm: Bool
3     var withLeg:           
4     init(          ){
5         self.withArm = withArm
6     }
7     init(withArm: Bool, withLeg: Bool){
8         self.withArm = withArm
9         self.withLeg = withLeg
10    }
11 }
12 let robot1 = Robot(withArm: true)
13 print(robot1.withArm)
14 let robot2 = Robot(withArm: true, withLeg: false)
15 print(          )
```

Robot	<input type="checkbox"/>
"true\n"	<input type="checkbox"/>
Robot	<input type="checkbox"/>
"false\n"	<input type="checkbox"/>



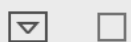
true
false

類別(class)DogRobot的型態為類別(class)Robot，繼承Robot的欄位變數。類別(class)DogRobot的override初始化方法，將覆蓋原繼承欄位的預設結果。請在第六行使用繼承類別的初始化方法，以繼承欄位變數，並在第七行設定繼承欄位變數的初始內容

```
1 class Robot{
2     var withArm = true
3 }
4 class DogRobot: Robot{
5     override init(){
6         super.           
7                    = false
8     }
9 }
10 let dogRobot = DogRobot()
11 print(dogRobot.withArm)
```



```
DogRobot
>false\n"
```



false

請將常數ant的型態宣告為Ant，並在第八行使用方法move

```
1 class Ant{
2     var position = (x:0, y:0)
3     func move(){
4         self.position.x += 1
5     }
6 }
7 let ant = 
8     ()
9 print(ant.position)
10
11
```

Ant

Ant

"(x: 1, y..."

(x: 1, y: 0)

類別(class)SupAnt的型態為Ant，關鍵字override func使得類別方法move覆蓋所繼承類別的move方法。

```
1 class Ant{
2     var position = (x:0, y:0)
3     func move(){
4         self.position.x += 1
5     }
6 }
7 class SupAnt: Ant{
8     override fun move(){
9         self.position.x += 10
10    }
11 }
12 let ant = SupAnt()
13 ant.move()
14 print(ant.position)
```

SupAnt

SupAnt

"(x: 10,..."

(x: 10, y: 0)

使用init進行類別型態Neuron的欄位變數初始化，本題的init方法有兩個輸入參數，參數名稱分為weight及activation，其中weight為具備Double實數之陣列，而activation的型態類別為列舉(enum) Activation。

在設定時，分別將self.weight及self.activation所代表的兩個欄位變數，指定為輸入參數weight及activation的內容

```
1 import UIKit
2 enum Activation{
3     case linear
4     case threshold
5     case tanh
6     case sigmoid
7 }
8 class Neuron{
9     var weight: [Double]
10    var activation: Activation
11    init( [redacted] ){
12        self.weight = weight
13        [redacted] = activation
14    }
15 }
```

請根據列印結果，在第十六行設定初始值

```
1 import UIKit
2 enum Activation{
3     case linear
4     case threshold
5     case tanh
6     case sigmoid
7 }
8 class Neuron{
9     var weight: [Double]
10    var activation: Activation
11    init( ) {
12        self.weight = weight
13        = activation
14    }
15 }
16 let neuron = Neuron(weight: [1, 2], activation:
17 print(neuron.weight)
```

Neuron

"[1.0, 2..."

[1.0, 2.0]
tanh