十年前的 OOP 期中 － 1

Johnny Wang 重出江湖

Part.1 True/False

1.[X] A variable of type boolean can be explicitly converted to that of type int.

不行，Java 裡面 boolean 就是只有 true/false 兩種值（C 裡面可以轉換成數字）

2.[O] An advantage of using the Unicode character set is that it easily handles languages other than English.

對^^連中文都可以

3.[X] Applets were designed to run as stand-alone applications.

不對，Applets 通常是在瀏覽器裡面跑的

4.[X] In Java, allocated memory that is no longer needed should be deallocated.

不用手動清掉，Java 會自動做 Garbage Collection。參考 http://openhome.cc/Gossip/JavaGossip-V1/GarbageCollection.htm

在 C++中，使用 new 配置的物件，必須使用 delete 來 清除物件，以釋放物件所佔據的記憶體空間，如果沒有進行這個動作，則若物件不斷的產生，記憶體就會不斷的被耗用，最後使得記憶體空間用盡，然而使用 delete 並不是那麼的簡單，如果不小心清除了尚被參考的物件，或是物件間共用的資源，則程式就會發生錯誤，小心的使用 new 與 delete，一直是 C ++中一個重要的課題。

在 Java 中，使用 new 配置的物件，基本上也是必須清除以回收記憶體空間的，但是您並不用特別關心這個問題，因為 Java 提供「**垃圾收集**」（Garbage collection） 機制，在適當的時候，Java 執行環境會自動檢查物件，看看是否有未被參考的物件，如果有的話就清除物件、回收物件所佔據的記憶體空間。

5.[X] Every Java program automatically Imports the java.util package.

沒有，不然我們就不用每次都在那邊 import java.util.scanner

6.[X] The equality operator (==) may be used to test if two string objects contain the same value.

不一定。真的要比較字串 A,B 的話應該要用 A.equals(B)。
因為 string 是物件，所以==的功用是判斷兩個物件的指標是否指到同樣的地方。
這時會發生一件有趣的事（http://goo.gl/FGhjzf）：

當使用 String literals(一串被雙引號括住的字元)時，使用 '=='運算子和使用 equals method 的結果會是一樣的。所有的 String literals 都是指向同一個 String 類別的 instances。系統中有一個 pool，當有新的 String literals 出現時，系統會先去檢查 pool 之中，是不是已經存在一個和這個新的 String literals 有相同內容的物件。如果存在，則會傳回一個指向這個此物件的 reference。若不存在，則會將此 String literals 放到 pool 中，然後傳回這個物件的 reference。

舉個例子：

```
String s1 = "hello";    String s2 = "hell"+"o";
System.out.println("Using equals op"+ (s1==s2)); //True
System.out.println("Using equals method" + (s1.equals(s2))); //True
```

當字串是由"new"這個關鍵字所造出來的時候，則不是這麼一回事。

```
String s3 = new String("hello");    String s4 = new String("hello");
System.out.println("Using equals op" + (s3==s4)); //False
System.out.println("Using equals method" + (s3.equals(s4))); //True
```

傳用"new"這個關鍵字時，會造出兩個不同的物件，所以會有兩個不同的 references，即使在底層的 string literal 是一樣的。在上面的例子中，'=='運算子傳回 false，因為兩個 references 是不同的。而 equals method 則傳回 true，因為這兩個物件所代表是同樣的字元序列。

7.[X] The three expressions at the start of a for statement are separated by two commas.

這有點在考英文 XD，comma 是逗號， semicolon 是引號（;）

8.[X] In a switch statement, the default case is always executed.

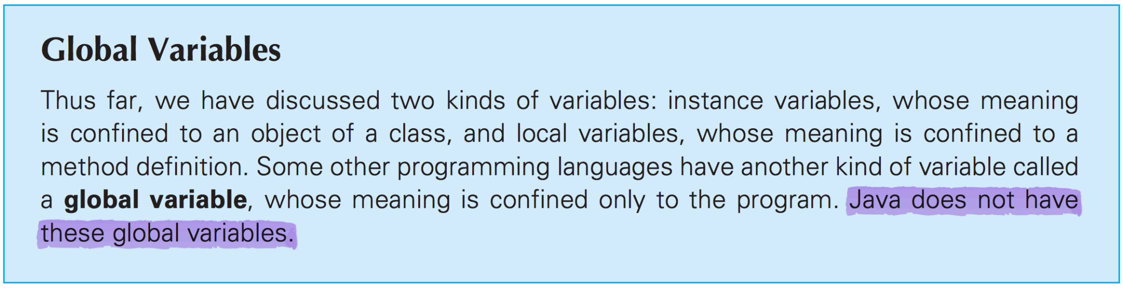
Switch 的 default case 是指如果都沒有符合上面的條件才執行 default case，跟 C 一樣

9.[O] An object of class A is an instance of class A.

空虛題

10.[X] The Java language supports global variables.

不不不哦，看課本 P.187：



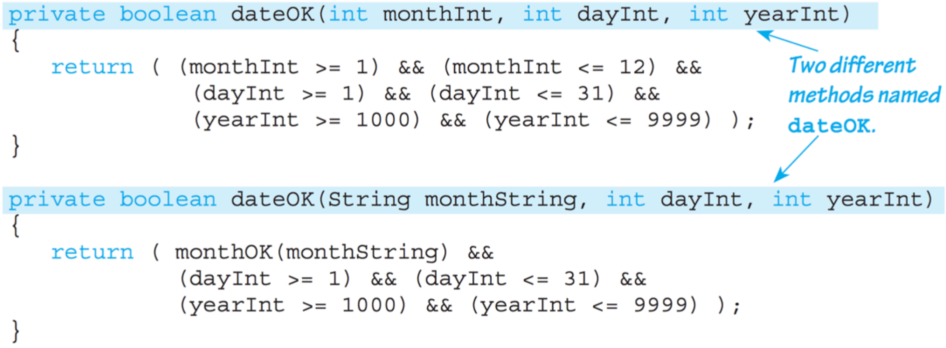
Global Variables

Thus far, we have discussed two kinds of variables: instance variables, whose meaning is confined to an object of a class, and local variables, whose meaning is confined to a method definition. Some other programming languages have another kind of variable called a **global variable**, whose meaning is confined only to the program. Java does not have these global variables.

11.[X] Java supports operator overloading.

這個是大陷阱 XD，Java 支援 overloading，但是不支援 operator overloading。

Overloading 是定義很多一樣名稱的函數，不一樣的 parameter。像這樣：

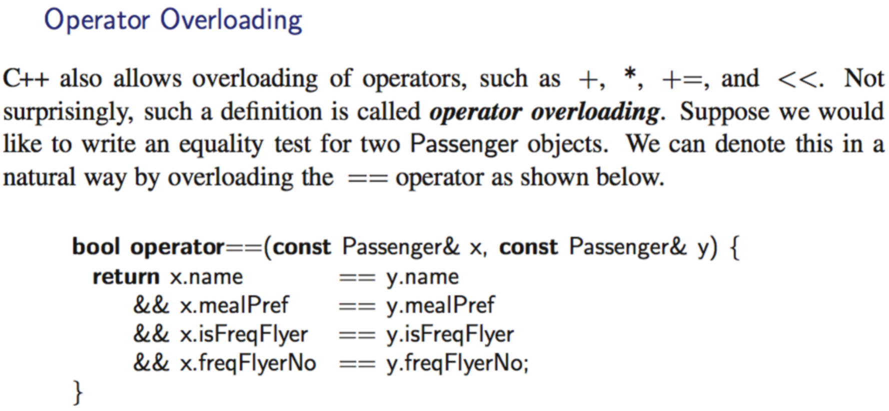


```
private boolean dateOK(int monthInt, int dayInt, int yearInt)
{
   return ( (monthInt >= 1) && (monthInt <= 12) &&
             (dayInt >= 1) && (dayInt <= 31) &&
             (yearInt >= 1000) && (yearInt <= 9999) );
}

private boolean dateOK(String monthString, int dayInt, int yearInt)
{
   return ( monthOK(monthString) &&
            (dayInt >= 1) && (dayInt <= 31) &&
            (yearInt >= 1000) && (yearInt <= 9999) );
}
```

Two different methods named `dateOK`*.*

Operator overloading 是你去把 class 的加減乘除定義出來，仿造設計函數的方式去做一個運算子的感覺，Java 沒有，C++有。(圖截自 C++資結課本)



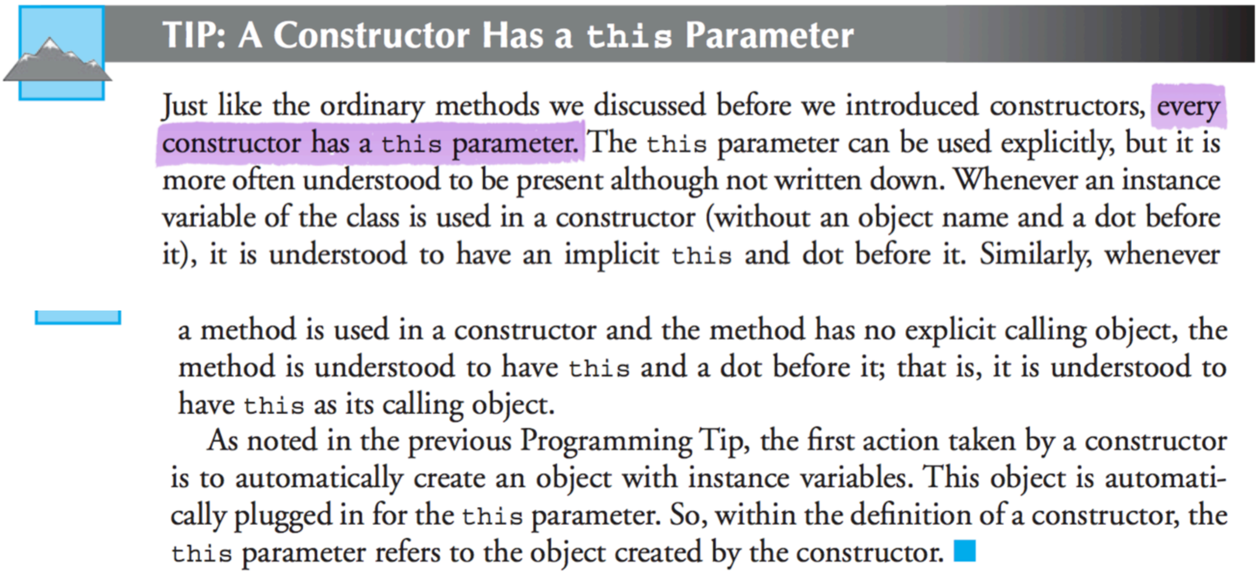
Operator Overloading

C++ also allows overloading of operators, such as +, *, +=, and <<. Not surprisingly, such a definition is called ***operator overloading***. Suppose we would like to write an equality test for two Passenger objects. We can denote this in a natural way by overloading the == operator as shown below.

```
bool operator==(const Passenger& x, const Passenger& y) {
  return x.name          == y.name
      && x.mealPref      == y.mealPref
      && x.isFreqFlyer   == y.isFreqFlyer
      && x.freqFlyerNo   == y.freqFlyerNo;
}
```

12.[　　] Only the default constructor has the this parameter

呃沒有，所有的都會傳 this，代表那個正在用的 object。看書比較快 XD:



TIP: A Constructor Has a `this` Parameter

Just like the ordinary methods we discussed before we introduced constructors, every constructor has a `this` parameter. The `this` parameter can be used explicitly, but it is more often understood to be present although not written down. Whenever an instance variable of the class is used in a constructor (without an object name and a dot before it), it is understood to have an implicit `this` and dot before it. Similarly, whenever a method is used in a constructor and the method has no explicit calling object, the method is understood to have `this` and a dot before it; that is, it is understood to have `this` as its calling object.

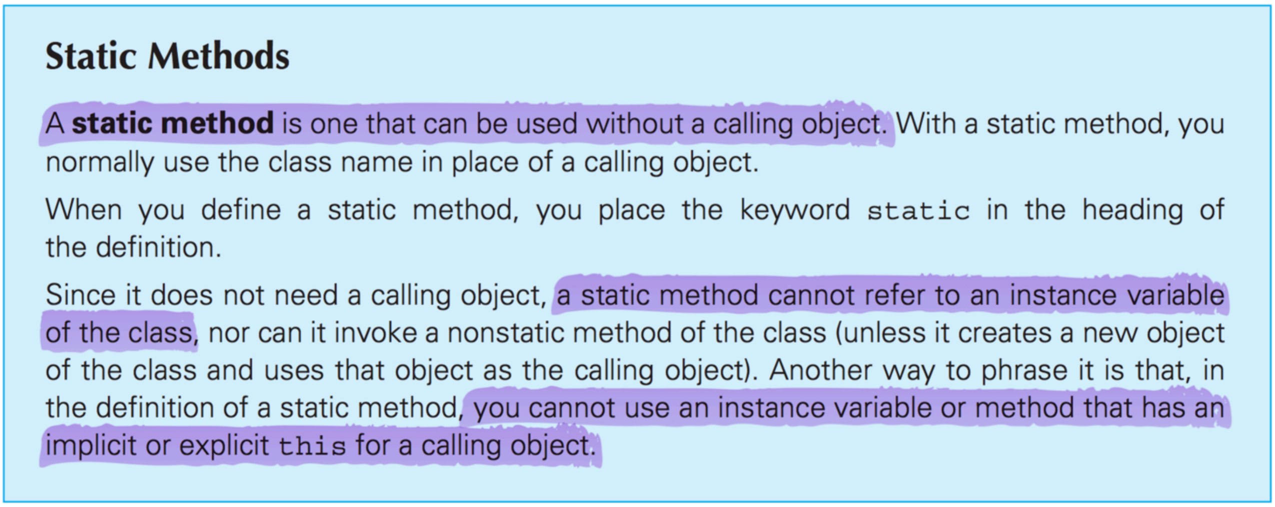
As noted in the previous Programming Tip, the first action taken by a constructor is to automatically create an object with instance variables. This object is automatically plugged in for the `this` parameter. So, within the definition of a constructor, the `this` parameter refers to the object created by the constructor. ■

13.[O] Because of pass-by-value, the passed contents of the object can be changed in the called method, but the original object refererence is never changed.

這題感覺上是無聊的兩層陷阱。假設我們把一個 A 物件傳進去 B 方法裡面。傳的時候我們是用 A 的 reference 在傳，所以我們在 B 方法裡面可以改 A 的內容，也確實會改到真正的那個 A object，但是這個 A object 在的位置一直都不會變。就算你把 A 的 reference 傳進去然後大改甚至換成另外一個 C object，因為 Java always pass by value 的關係，所以那個 reference 不會被動到。(個人認為藍色部分不需要懂啦 XD)

14.[X] In a static method, you may use the this parameter either exploitly or implicitly.

基本上是不行的。Static Method 就是那種很獨立的函數（但他還是被定義在某個 class 裡面）。所以我們會用 someclass.someStaticMethod(xxx,xxx)去呼叫他（用 class 名稱+點，不是用 object 名稱+點哦），但是在這個 someStaticMethod 裡面你是不能用 this 去叫 someclass 的。



Static Methods

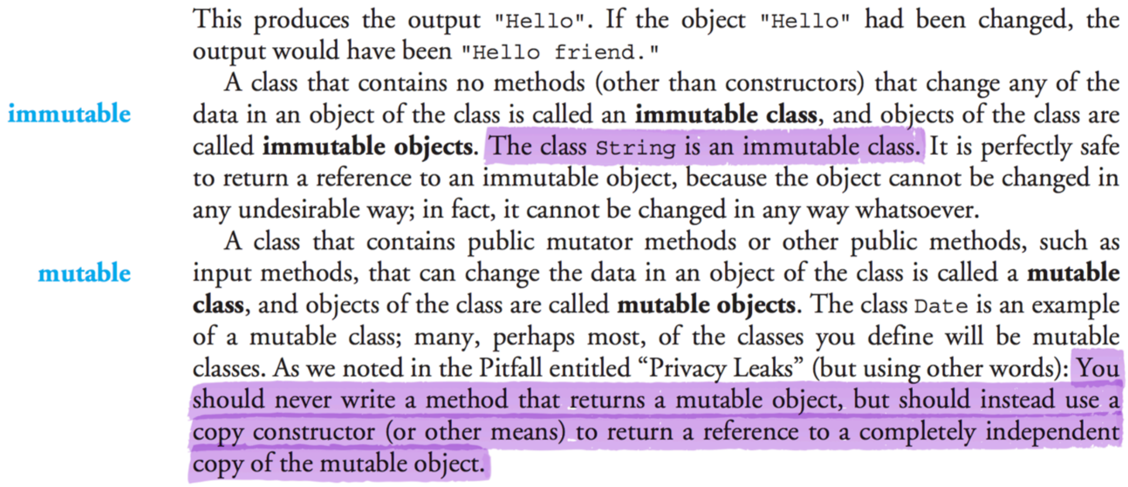
A **static method** is one that can be used without a calling object. With a static method, you normally use the class name in place of a calling object.

When you define a static method, you place the keyword `static` in the heading of the definition.

Since it does not need a calling object, a static method cannot refer to an instance variable of the class, nor can it invoke a nonstatic method of the class (unless it creates a new object of the class and uses that object as the calling object). Another way to phrase it is that, in the definition of a static method, you cannot use an instance variable or method that has an implicit or explicit `this` for a calling object.

15.[X] The String class is a mutable class.

這有點微難懂。先看課本：



This produces the output `"Hello"`. If the object `"Hello"` had been changed, the output would have been `"Hello friend."`

immutable

A class that contains no methods (other than constructors) that change any of the data in an object of the class is called an **immutable class**, and objects of the class are called **immutable objects**. The class `String` is an immutable class. It is perfectly safe to return a reference to an immutable object, because the object cannot be changed in any undesirable way; in fact, it cannot be changed in any way whatsoever.

mutable

A class that contains public mutator methods or other public methods, such as input methods, that can change the data in an object of the class is called a **mutable class**, and objects of the class are called **mutable objects**. The class `Date` is an example of a mutable class; many, perhaps most, of the classes you define will be mutable classes. As we noted in the Pitfall entitled "Privacy Leaks" (but using other words): You should never write a method that returns a mutable object, but should instead use a copy constructor (or other means) to return a reference to a completely independent copy of the mutable object.

immutable 的意思就是不能隨便改，String 是那種你不能改裡面內容的物件。如果有一個 String A 的內容是"hahaha"，我們今天要把 A 的內容改成 hahahahahaha 的時候，java 的作法是把原本放 hahaha 的空間放棄，另外找一個空間放 hahahahahaha，而不用改內容的方式。

Part.2 Choices

1.[a] Identify the invalid Java identifier:

(a)1Week (b)Week1 (c)amountDue (d)amount_due

命名規則，第一個字不能是數字

2.[d] In Java, source code is compiled into object code called ___________.

(a)Bit-code (b)Class code (c)Method code (d)Byte-code

執行流程，java compiler 先把 source code 轉乘 byte-code，再由 Java Virtual Machine 去執行 Byte-Code

3.[b] The value of the expression (int)27.6 evaluates to:

(a)28 (b)27 (c)26 (d)None of the above.

BJ4

4.[b] What is the value of the variable c in the statements that follow?

String phrase = “Make hay while the sun is shining.”;

Char c = phrase.charAt(10);

(a)w (b)h (c)i (d)None of the above.

從 M 是[0]開始數

5.[ ] To make a block comment for inclustion in the Javadoc documentation, the block must be delimited by:

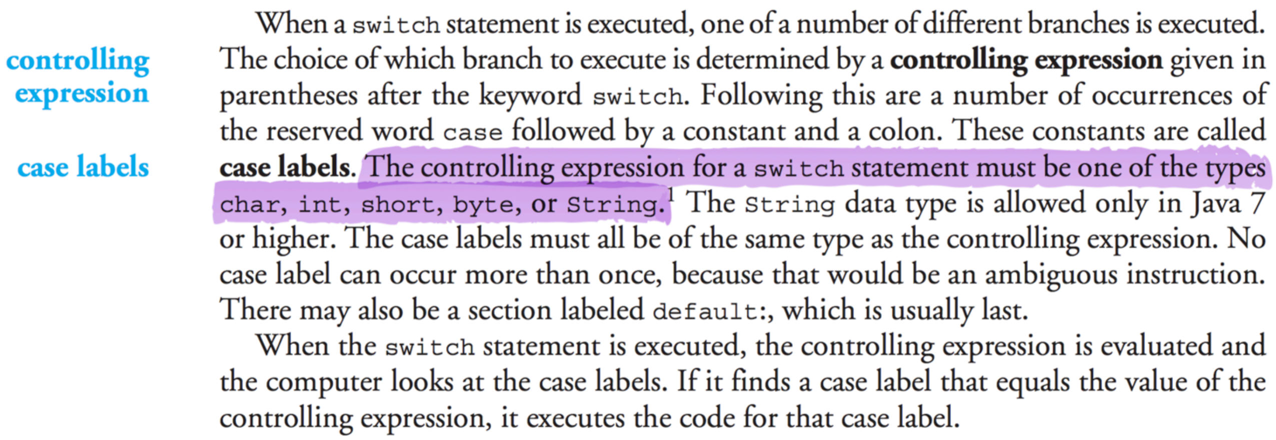
(a)/** / (b)*/* / (c)**/ /* (d)**/ */

javadoc 的規定。Javadoc 是一種如果程式註解照他的規定寫，java 會自動幫你產生說明文件的溫馨東西。課本 p.330

6.[d] The controlling expression for a switch statement includes all of the following type except:

(a)char (b)int (c)byte (d)double

這個意思是說在 switch 裡面會有一個要被判斷的東西，那個只能是 char,int,short,byte,String。



controlling expression

case labels

When a `switch` statement is executed, one of a number of different branches is executed. The choice of which branch to execute is determined by a **controlling expression** given in parentheses after the keyword `switch`. Following this are a number of occurrences of the reserved word `case` followed by a constant and a colon. These constants are called **case labels**. The controlling expression for a `switch` statement must be one of the types `char`, `int`, `short`, `byte`, or `String`.[1] The `String` data type is allowed only in Java 7 or higher. The case labels must all be of the same type as the controlling expression. No case label can occur more than once, because that would be an ambiguous instruction. There may also be a section labeled `default:`, which is usually last.

When the `switch` statement is executed, the controlling expression is evaluated and the computer looks at the case labels. If it finds a case label that equals the value of the controlling expression, it executes the code for that case label.

7.[b] The looping mechanism that always executes at least once is the _________ statement.

(a)if…else (b)do…while (c)while (d)for

do{…}while(condition);一定會先執行一次{…}的東西

8.[b] A ________ statement terminates the current iteration of a loop.

(a)Break (b)Continue (c)Switch (d)Assert

Break 是跳掉整個迴圈，Continue 是直接結束這回合(XD)，會進入下一回合的 loop。

9.[a] When using a compound boolean expression joined by an &&(AND) in an if statement:

(a)Both expressions must evaluate to true for the statement to execute.

(b)The first expression must evaluate to true and the second expression must evaluate to false for the statement to execute.

(c)The first expression must evaluate to false and the second expression must evaluate to true for the statement to execute.

(d)Both expressions must evaluate to false for the statement to execute.

打這麼多字結果這題這麼無聊= =

10.[c] Java has a way of officially hiding details of a class definition. To hide details, you mark them as _____.

(a)public (b)protected (c)private (d)all of above

呃這題也是有點無聊，private 就是外面 class 不能存取

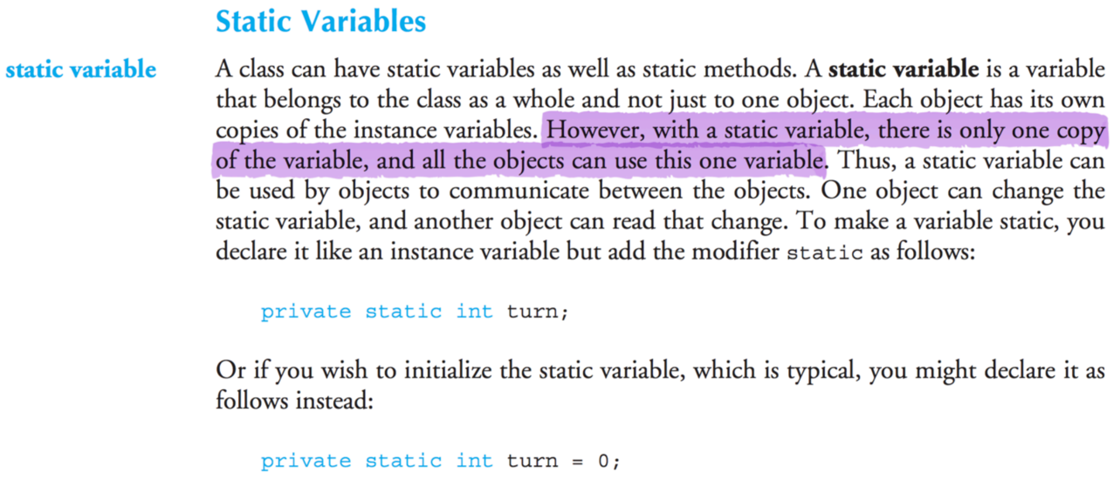
11.[a] The name of a method and the list of _____ type in the heading of the method definition is called the method signature. (a)parameter (b)argument (c)return (d)primitive

是 parameter。和 argument 的差異是：parameter 是定義那邊的東西，argument 是真正用的時候的東西。例如：

```
public static int addTwo(int a,int b){   //這裡 a,b 是 parameter
        return a+b;
}
int c = addTwo(1,2);   //這裡 1,2 是 argument
```

12.[a] Only _____ copy/copies of a static variable are available to objects of a class.

(a)one (b)two (c)three (d)none of above.



Static Variables

static variable

A class can have static variables as well as static methods. A **static variable** is a variable that belongs to the class as a whole and not just to one object. Each object has its own copies of the instance variables. However, with a static variable, there is only one copy of the variable, and all the objects can use this one variable. Thus, a static variable can be used by objects to communicate between the objects. One object can change the static variable, and another object can read that change. To make a variable static, you declare it like an instance variable but add the modifier `static` as follows:

```
private static int turn;
```

Or if you wish to initialize the static variable, which is typical, you might declare it as follows instead:

```
private static int turn = 0;
```

這裡的 static variable 的意思是說整個 class 的每一個 object 都會用那個同樣的一個變數。可以作為同 class 的不同 object 溝通的方式。

13.[a] All of the following are wrapper classes except:

(a)String (b)Integer (c)Character (d)Double

wrapper classes 的意思是說，原本這些基本資料型態(primitive type)可以藉由像是包裝(wrap)的方式，把它包成物件，例如把 int 包成 Integer 這種物件。阿原本就是物件的 String 就不是。

14.[d] When you use the assignment operator with variables of a class type, you are assigning a:

(a)value (b)primitive type (c)local variable (d)reference.

拿 object 來做 assign 的時候不會複製整個 object，只會做指標的參考。例如 ClassA a = new ClassA(); ClassA b = a;那 b 只是獲得了 a 的 reference

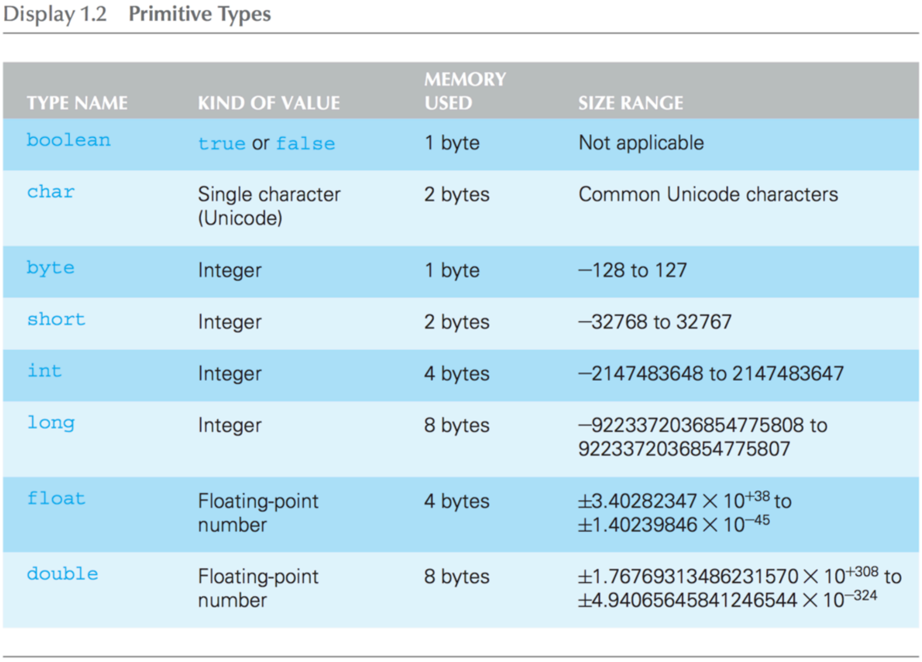
15.[b] A condition that allows a programmer to circumvent the private modifier and change the private instance variable is called:

(a)a copy constructor (b)a privacy leak (c)a class invariant (d)an anonymous object

會有 privacy leak 通常是使用一些邏輯正確但不安全的寫法。可以看問答最後一題。

Part.3 Short Answer: (40%)

1. List the primitive data tpyes Java supports. Indicate the number of bytes each type used.



Display 1.2 Primitive Types

TYPE NAME	KIND OF VALUE	MEMORY USED	SIZE RANGE
boolean	true or false	1 byte	Not applicable
char	Single character (Unicode)	2 bytes	Common Unicode characters
byte	Integer	1 byte	−128 to 127
short	Integer	2 bytes	−32768 to 32767
int	Integer	4 bytes	−2147483648 to 2147483647
long	Integer	8 bytes	−9223372036854775808 to 9223372036854775807
float	Floating-point number	4 bytes	$\pm 3.40282347 \times 10^{+38}$ to $\pm 1.40239846 \times 10^{-45}$
double	Floating-point number	8 bytes	$\pm 1.76769313486231570 \times 10^{+308}$ to $\pm 4.94065645841246544 \times 10^{-324}$

2. Describe the detail construction process of the following statements:

Integer tmp = new Integer(19);

目前想法：(求高手救援)

1. 先右邊：new Integer(19)會去執行 Integer(int)這個 constructor，建立一個放在 heap 的物件
2. 左邊宣告一個 tmp 是 Integer type 的 reference，這個 reference 會放在 stack，裡面的內容指向剛剛宣告放在 heap 的物件

3. Correct the following code:

```
public class EX{
        public static void main(String[] args){
        long x = 0L;
                for(x=1;x<20;x++){
                        switch(int(x%2)){
                        case 1: System.out.println(x+" is odd");break;
                case 0: System.out.println(x+” is even”);break;
                }
            }
        }
}
```

4. Correct the following constructor without changing the logic.

```
public class Order{
    private String acct = “AAA”;
    private String item = “pen”;
    private double price = 25.0;
    private int qty = 1;
    public Order (String acct,String item, double price, int qty){
        this.acct = acct;
        this.item = item;
        this.price = (price<0) ? 0 : price;
        this.qty = (price<0) ? 0 : qty;
    }
}
```

5. In order to avoid the privacy leakage of class A, please correct the following statement.
(hint: rewrite the accessor and mutator of attributes math with clone constructor of class Credit)

```
public class Student{
    private Credit math = new Credit();
    public Credit getCredit(){   return new Credit(math) ; }
    public void setCredit (Credit tmp) {   math = new Credit(tmp) ; }
}

public class Credit{
    private int score;
    public Credit(Credit tmp){
        score = tmp.score;
    }
}
```