＊程式設計方法的演進：

程序導向(procedure-oriented)－

1. 自訂函數：必須有一個回傳值。
2. 自訂程序：無回傳值。
3. 合併以自訂函數來表達自定程序

VB: Function Get(x,y) As Integer …GetData=xxxx end Function

C: int getdata(int x, int y)...return xxx;

void setdata()…無回傳值

＊物件導向(object-oriented)－

1.類別(class)：藍圖、企畫書

2.類別的成員(memeber)：屬性、方法/成員方法、建構子

* class Human {

int head = 1;

int eyes = 2;

int mouth = 1;

Human(){random}

Human(String gender){}

void run(int distance){}

void eat(String food){}

}

＊建立第一個物件

Human Adam = new Human();

Stack: Adam

Heap: head, eyes, mouth

Static:

Human Eva;

Eva = new Human(“female”);

Class Test {

static int x;

int y;

void show(){

int z;

}

}

Test.java => Test.class

載入class叫用x：

Test text = new Test();

＊輸入字元：鍵盤→buffer→程式

char ch = (char)Ststem.in.read();

byte → byte

short → Short

int → Integer

long → Long

float → Float

double → Double

char → Character

boolean → Boolean

int x = Integer.parseInt(“123”);

int y = 456;

Stirng s = Integer.toString(456);

使用者輸入password，如何針對password的內容是否為”abcd”

If(password.equals(“abcd”))

System.out.print(“Welcome!”);

else{

System.err.println(“please check your password again.”);

System.exit(0);

}

物件(實體)的比較 1. == 比較是否為同一個物件 2.equals()比較內容是否相同

compareTo(): 比較內容的大小

1. Byte, Short－回傳數值的差值
2. Integer, Long, Float, Double－回傳-1,0,1
3. Character－回傳unicode的差值
4. Boolean－回傳true與false

父類別 物件 = new 子類別()

Polymorphism

Class A { }

Class B : A

A obj = new B();