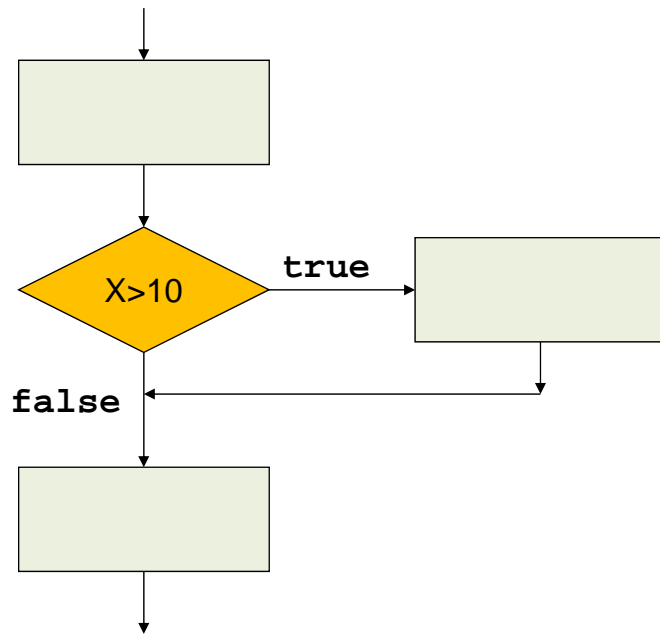
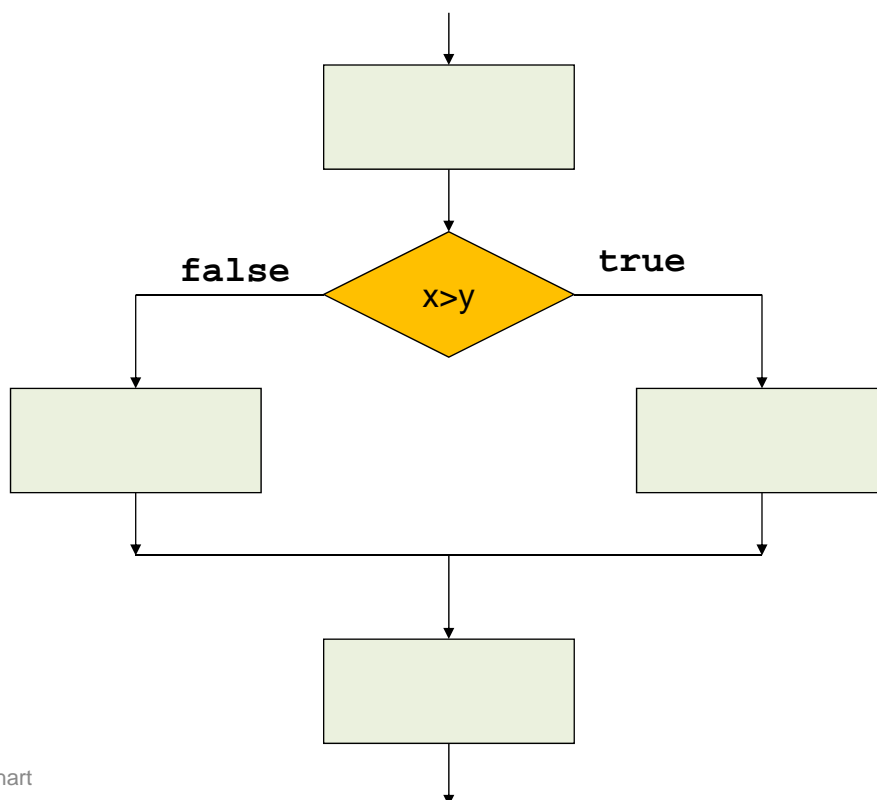


流程圖

if 流程



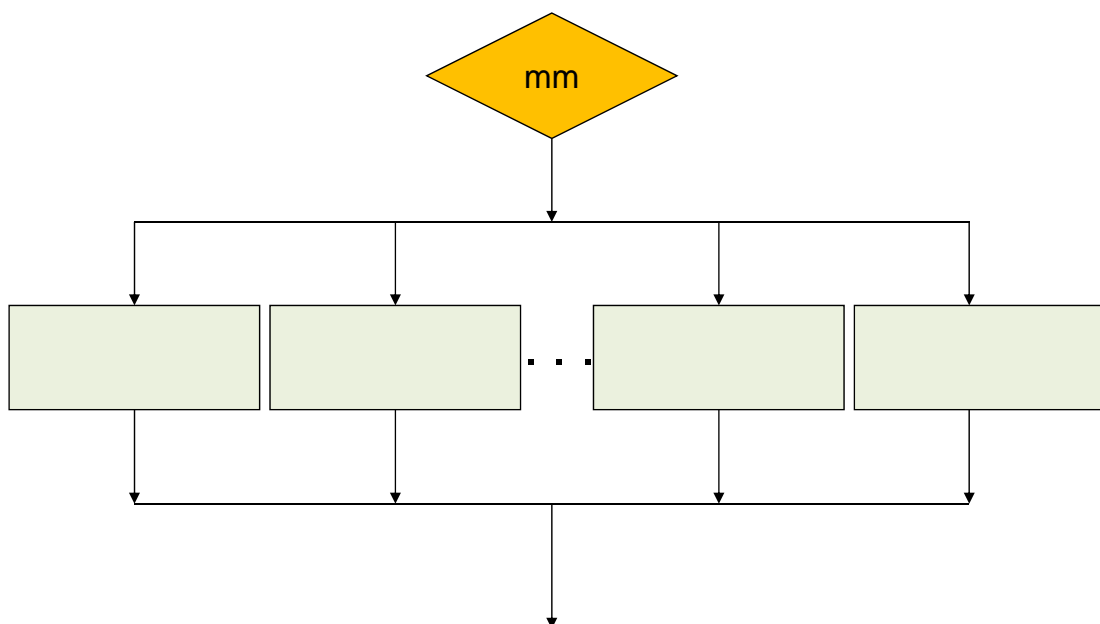
if-else 流程



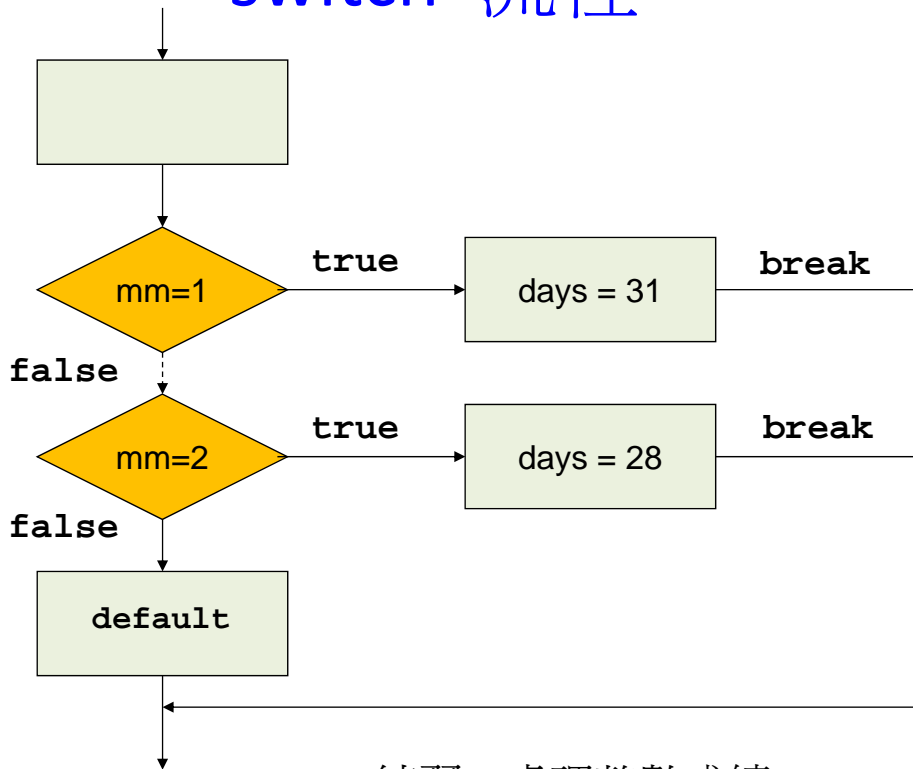
練習

- 寫一個程式，輸入西元年份 yy ，輸出該年是否閏年
- yy is divisible by 400
- yy is divisible by 4 but (not divisible by 100)

Switch-Case 概念



switch 流程

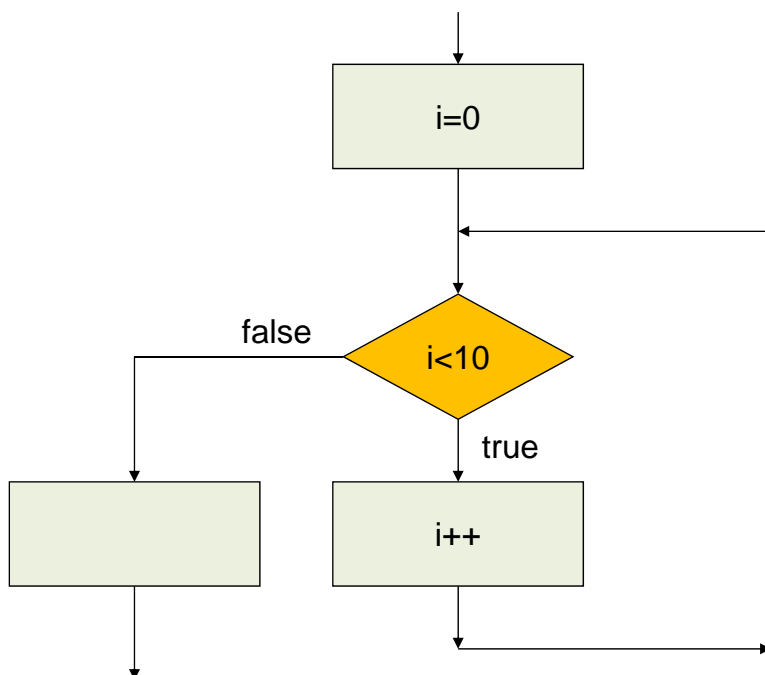


Raptor flow chart

練習：處理整數成績

5

for 流程

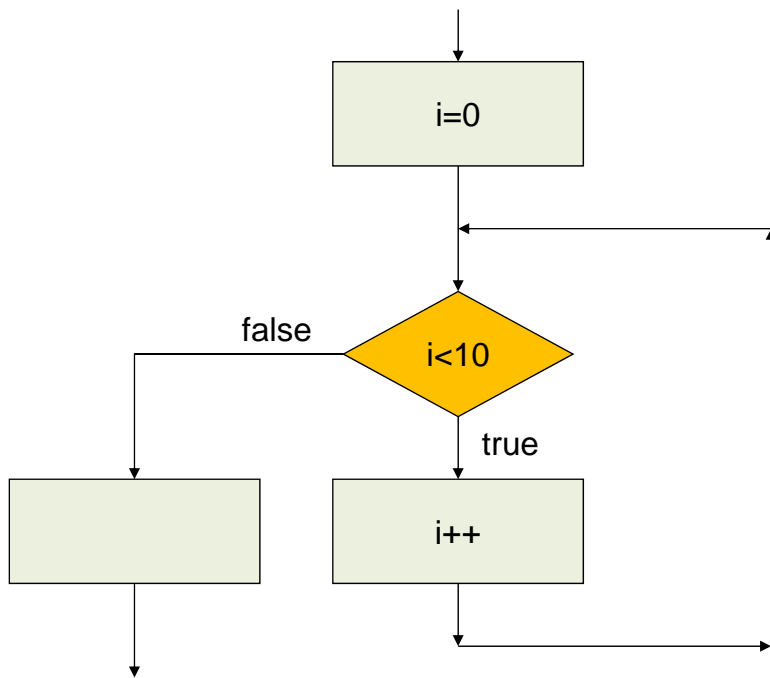


Raptor flow chart

練習：計算1到100的偶數和

6

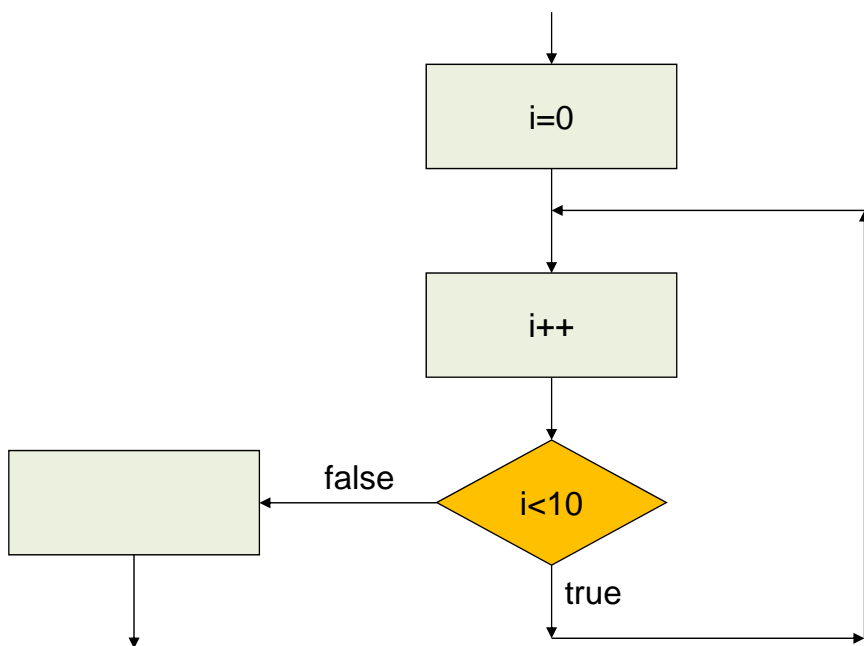
while 流程



Raptor flow chart

7

do-while 流程



Raptor flow chart

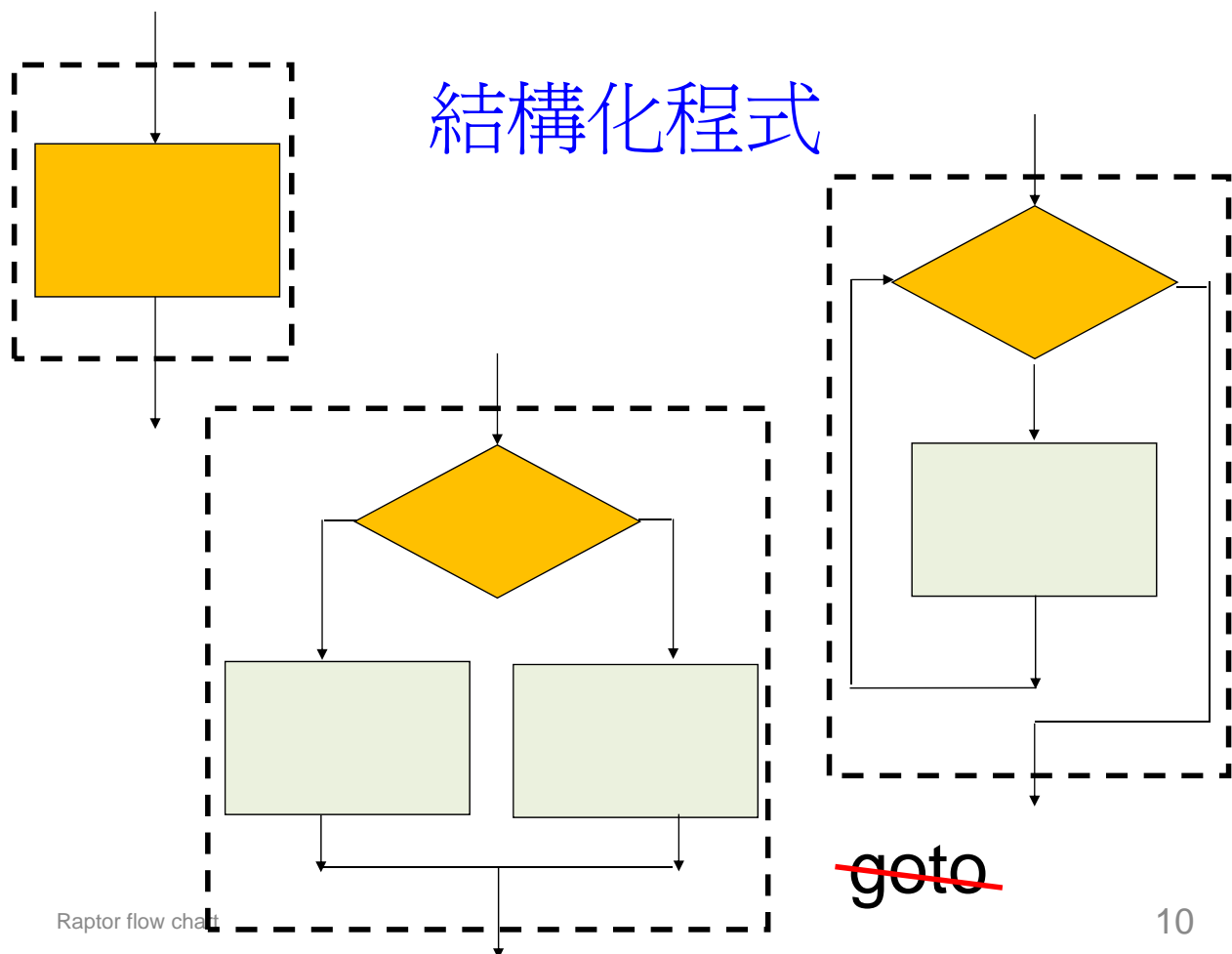
8

關於迴圈(Loops)

- 迴圈之基本要素
 - 控制變數初值設定
 - 檢驗條件
 - 控制變數改變
- 三種迴圈敘述之不同
- 常見錯誤
 - Off-by-one (少一次)
 - 無窮迴圈
- 迴圈與執行效率
- 練習
- 寫一程式，要求使用者不斷輸入及Echo學生成績，直到輸入為負時為止。印出學生總人數及平均成績

Raptor flow chart

9



Raptor flow chart

10

虛擬/偽代碼(Pseudo Code)

- 幫助思考程式流程
- 沒有程式語言正式，省略細節，易寫易懂
- 限定使用結構化程式流程控制，容易以程式語言改寫
- 較流程圖方便
- 紙筆追蹤測試(Tracing an algorithm)

Euclid 輾轉相除法

輸入正整數 a & b

只要 ($b \neq 0$) {

$r = a \div b$ 後的餘數;

$a = b;$

$b = r;$

}

$gcd = a;$

測試數據

1: $a=12, b=56$

2: $a=120, b=18$

3: $a=105, b=13$

4: $a=356, b=220$

5: $a=2136, b=3312$

練習： $1+2+\dots+n \leq \text{limit}$

練習: 判斷是否質數的篩檢法

- 要判斷 x 是否質數，
可用 d ($2 \dots \sqrt{x}$)，逐一檢查是否能整除 x
- 如果都不能整除， x 便是質數，
否則 x 就不是質數
- 函式isPrime(x)
- 印出1到100的所有質數

Raptor flow chart

13

Raptor: Flowchart programming

Adding block / symbols

輸出

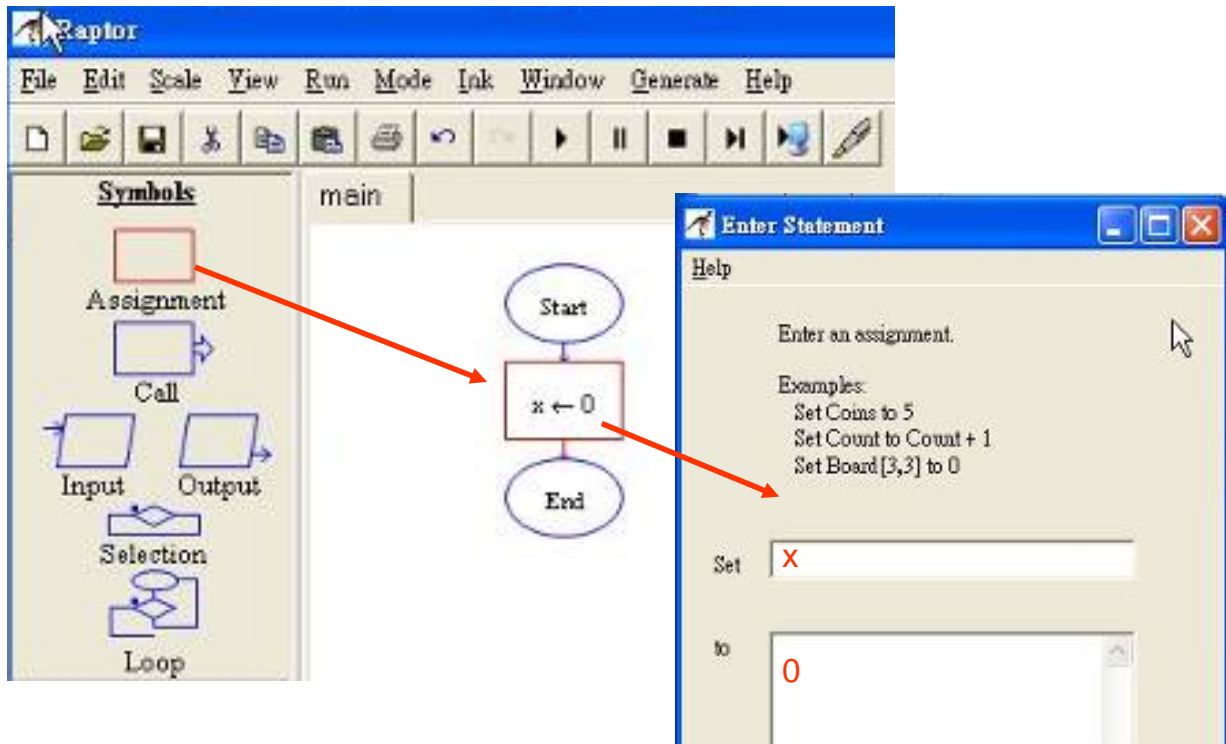
變數

Clear

Raptor Flowchart

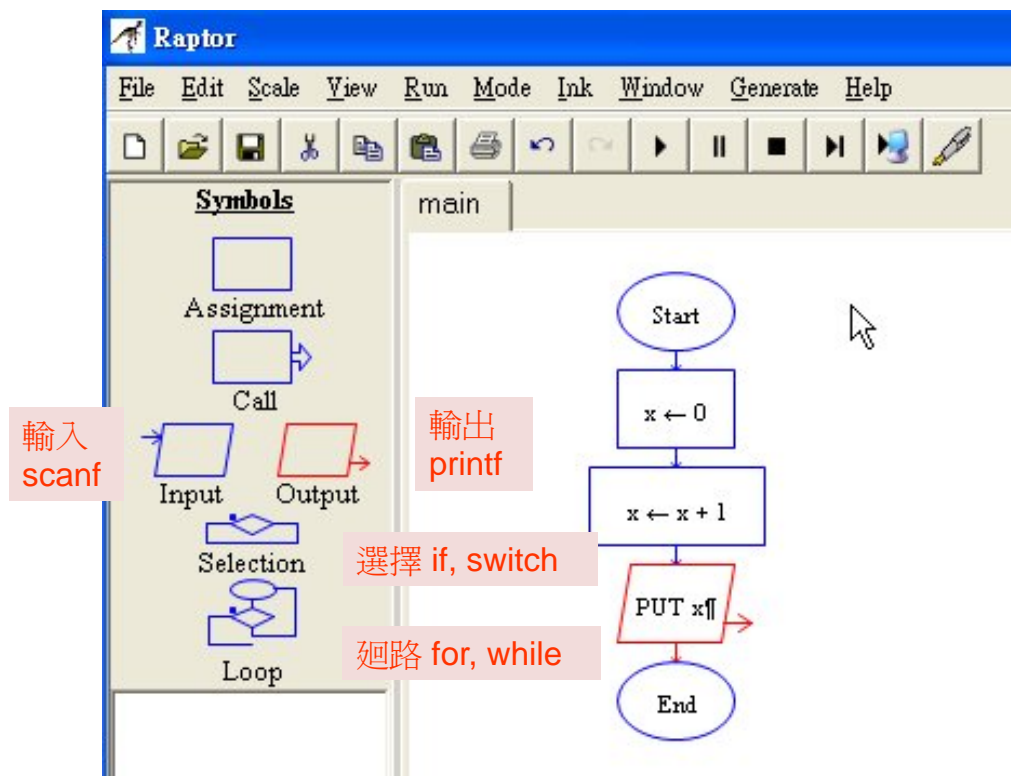
14

Enter an statement



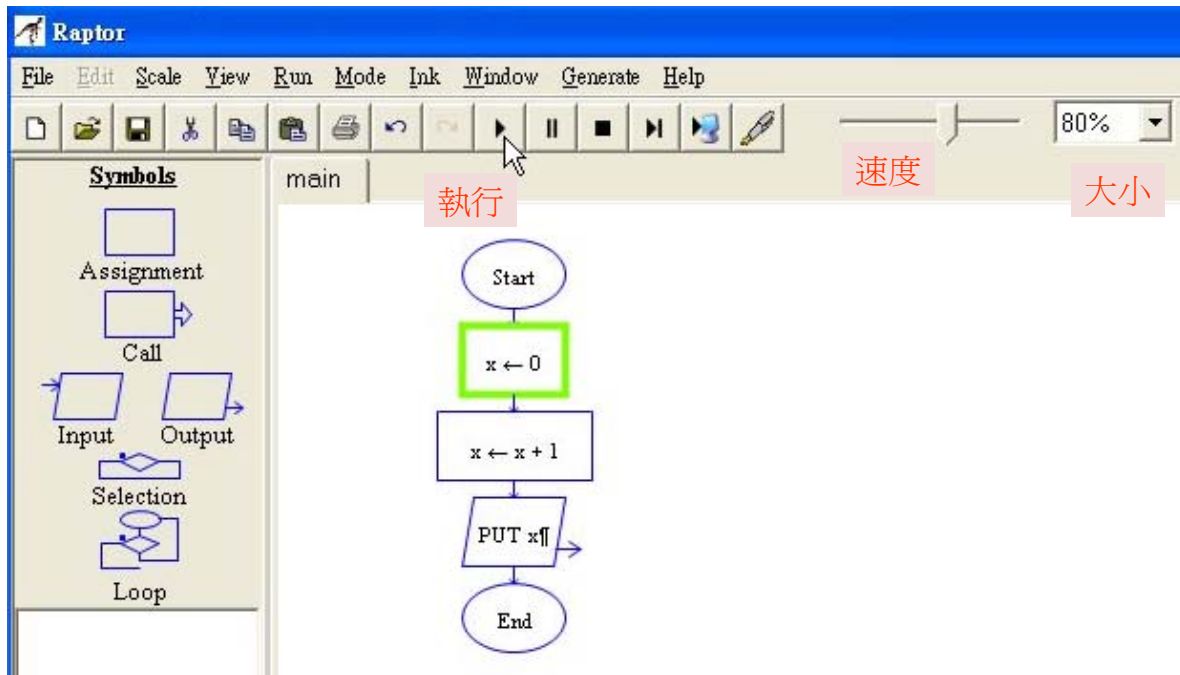
Raptor Flowchart

Create output symbol



Raptor Flowchart

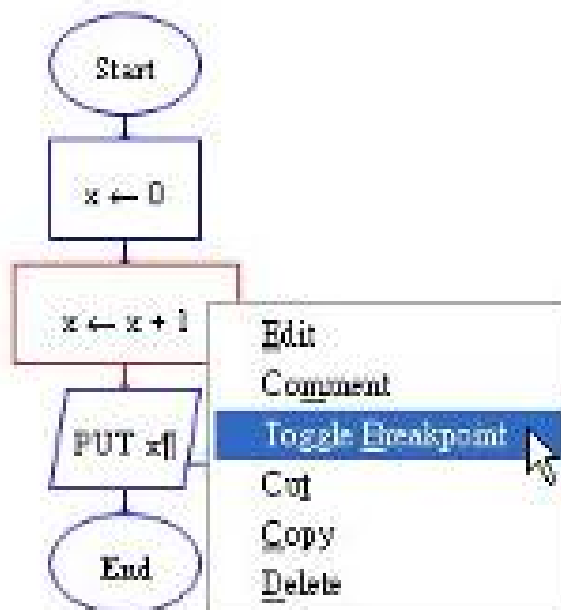
Running



Raptor Flowchart

17

Add Toggle breakpoint



Raptor Flowchart

18

Generate 3GL (C++/C#)

The screenshot shows the Raptor IDE interface. On the left is a 'Symbols' palette with icons for Assignment, Call, Input, Output, Selection, and Loop. The main workspace displays a flowchart for a simple program: Start -> x ← 0 -> x ← x + 1 -> PUT x -> End. A 'Generate' menu is open, showing options: Ada, C++, C#, IT105 Java, and Standalone. A text label 'Generate executable file' is placed near the menu. On the right, a window titled 'simple1.cpp - 記事4' displays the following C++ code:

```

#include <iostream>
#include <string>

using namespace std;
int main()
{
    ?? x;

    x = 0;
    x = x + 1;
    cout << x << endl;
    return 0;
}

```

Raptor Flowchart

The screenshot shows the Raptor IDE with a flowchart for a 'Digital Clock' exercise. The flowchart starts with 'Start', followed by 'mark ← 0' (labeled '處理框'), then an 'X Loop' (labeled '判定框'). Inside the loop, there is a decision diamond 'mark < 0'. The 'Yes' path leads to 'PUT fail', and the 'No' path leads to '分數(0-100) GET mark' (labeled '輸入/出'). This is followed by another decision diamond 'mark < 50'. The 'Yes' path leads to 'PUT pass', and the 'No' path loops back to the 'X Loop' (labeled '迭代 while, for'). The flowchart ends at 'End'. A text box at the bottom left contains the text '練習：Digital Clock'. The Raptor IDE interface includes a 'Run' button circled in red and a zoom slider set to 100%.

Raptor flow chart

Graphic window opening and closing procedures

Open_Graph_Window(X_Size, Y_Size)
Close_Graph_Window
Set_Window_Title("Title")

Graphic window "size" functions

Get_Max_Width
Get_Max_Height
Get_Window_Width
Get_Window_Height

Drawing procedures

Put_Pixel(X, Y, Color)
Draw_Line(X1, Y1, X2, Y2, Color)
Draw_Box(X1, Y1, X2, Y2, Color, Filled/Unfilled)
Draw_Circle(X, Y, Radius, Color, Filled/Unfilled)
Draw_Ellipse(X1, Y1, X2, Y2, Color, Filled/Unfilled)
Draw_Arc(X1, Y1, X2, Y2, StartX, StartY, EndX, EndY, Color)
Clear_Window(Color)
Flood_Fill(X, Y, Color)
Display_Text(X, Y, String Expression, Color)
Display_Number(X, Y, Number Expression, Color)

Freeze_Graph_Window
Update_Graph_Window
UnFreeze_Graph_Window

Raptor flow chart

21

procedure call

Delay_for(0.2) // 0.2 sec
Clear_Console
Draw_Circle(X, Y, 7, Blue)

Mouse input procedures

Wait_for_Mouse_Button(Button)
Get_Mouse_Button(Button, X, Y)

Mouse input functions

Mouse_Button_Pressed(Button)
Mouse_Button_Released(Button)
Get_Mouse_X
Get_Mouse_Y

Keyboard input procedure

Wait_For_Key

Keyboard input functions

Key_Hit
Get_Key // ASCII
Get_Key_String

Is_Open

To_Character (n)
To_ASCII (ch)

Redirect_Input ("file.txt")
Redirect_Input (True)
Redirect_Input (False)

Redirect_Output ("file.txt")
Redirect_Output (False)

Raptor flow chart

22

Function

Basic math:

sqrt, log, abs, ceiling, floor

Trigonometry:

sin, cos, tan, cot, arcsin,
arccos, arctan, arccot

Miscellaneous:

Length_Of, Random

Draw_Bitmap(Load_Bitmap("abc.bmp") ,x,y,w,h)

Mouse input procedures

Wait_for_Mouse_Button(Button)
Get_Mouse_Button(Button, X, Y)

Mouse input functions

Mouse_Button_Pressed(Button)
Mouse_Button_Released(Button)
Get_Mouse_X
Get_Mouse_Y

Sound

Play_Sound(wav)
Play_Sound_Background("snd.wav")
Play_Sound_Background_Loop("snd.wav")

Graphics window query function

color = Get_Pixel (X, Y)
Set_Font_Size(Size)
Get_Font_Height
Get_Font_Width
Random_Color //0-15
Random_Extended_Color //0-241
Closest_Color (R,G,B) //0-255
Draw_Bitmap (Bitmap, X, Y, W, H)

Get_Pixel

Returns

0 for Black,
1 for Blue,
...,
16 for White

RAPTORGraph Colors

Black, Blue, Green, Cyan, Red, Magenta, Brown,
Light_Gray, Dark_Gray, Light_Blue, Light_Green,
Light_Cyan, Light_Red, Light_Magenta, Yellow, White

How to animate an object in RAPTORGraph

Place the following inside of a loop

Draw an object relative to an X,Y point with the drawing procedures
Delay_For some small time period
Draw the object again in white (i.e. erase it)
Update the X,Y point where you are drawing by some small offset