

# BRIEFING SESSION ON NSS ICT ELECTIVE – Software Development

Centre for Information Technology in  
Education, HKU

# Elective Option D: Software Development

- Topic A on “Programming” (47 hours)
  - Choose a programming language from Pascal, C, Visual Basic, Java
- Topic B on “Programming Languages” (12 hours)
  - Programming paradigms
  - Language translators and compilers



# Topic A

Programming

# Topic A: Programming

A1	Getting Familiar with the Programming Environment	3 periods	2 hours
A2	Input and Output	3 periods	2 hours
A3	Arithmetic Operations and Functions	4½ periods	3 hours
A4	Selection Structure	6 periods	4 hours
A5	Iteration Structure	7½ periods	5 hours
A6	Modular Programming	9 periods	6 hours
A7	Arrays	6 periods	4 hours
A8	Strings	4½ periods	3 hours
A9	File Handling	6 periods	4 hours
A10	Searching, Sorting and Merging	10½ periods	7 hours
A11	Stacks, Queues and Linked Lists	10½ periods	7 hours
		Total:	<b>70½ periods</b>
			<b>47 hours</b>

\* 1 period = 40 minutes

# Resource package for Topic A

- 11 Chapters (A1 – A11)
- 4 Learning tasks
  - A1 – A3: Learning Task A #1
  - A4 – A6: Learning Task A #2
  - A7 – A9: Learning Task A #3
  - A10 – A11: Learning Task A #4
- 4 Assessment tasks
  - A1 – A3: Assessment Task A #1
  - A4 – A6: Assessment Task A #2
  - A7 – A9: Assessment Task A #3
  - A10 – A11: Assessment Task A #4
- Teaching plans for each chapter
- Source programs of examples and practical tasks
- List of references

# Resource package for Topic B

- 2 Chapters (B1 – B2)
- 2 Learning tasks
- 2 Assessment tasks
- Teaching plans for each chapter
- List of references

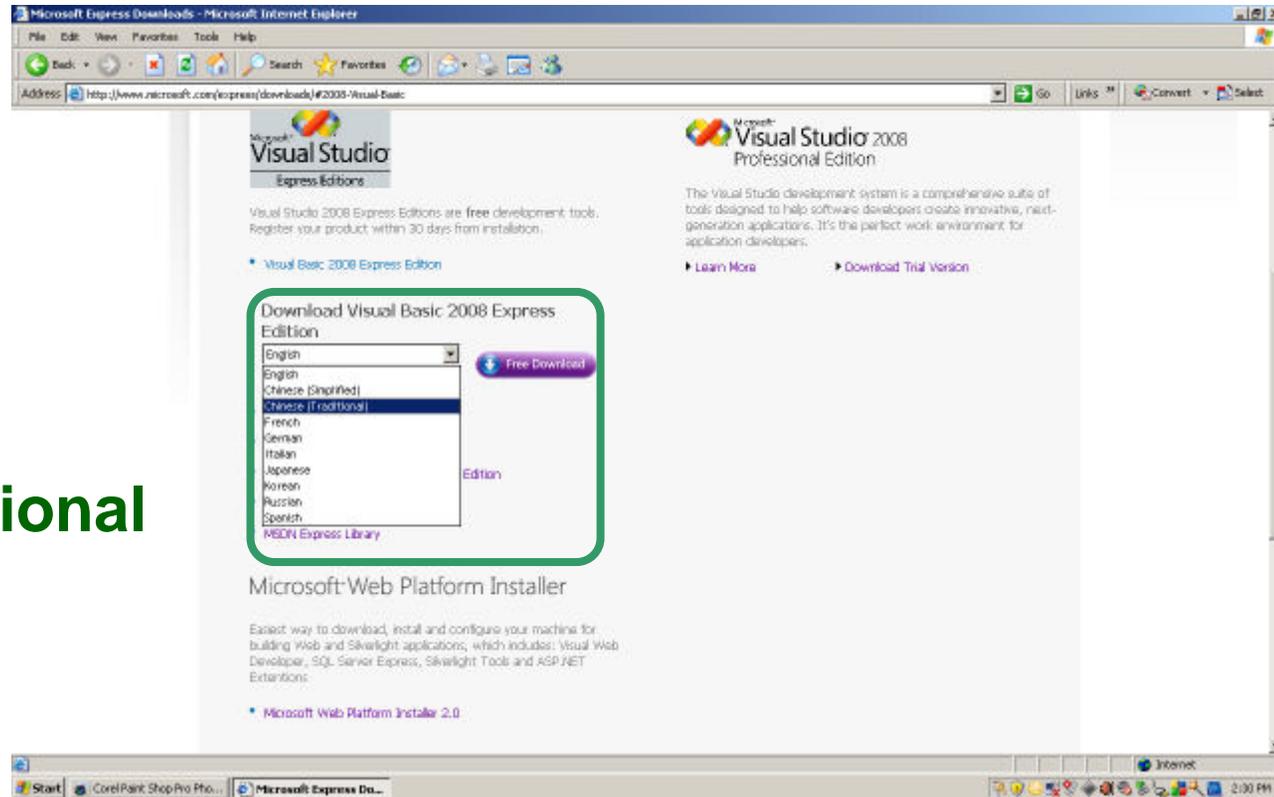
# Highlight on Visual Basic

- Pascal, C, Visual Basic, Java
- More resources available for Pascal and C
- Highlights on **Visual Basic** and **Java**

# Highlight on Visual Basic

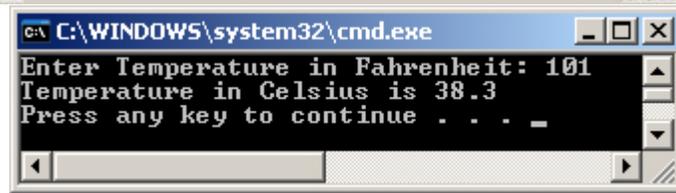
- Visual Basic 2008 Express

- <http://www.microsoft.com/express/Downloads/#2008-Visual-Basic>



- Free version
- English
- Chinese Traditional

- Console application



# Minimalist approach

- Students start as soon as possible on meaningful tasks
  - Understand the instructions with a not-so-steep learning curve
  - Use of VB **Immediate Window**

## Immediate Window

```
length = 3
width = 4
? length * width
12 {Integer}
    Integer: 12 {Integer}
```

## Immediate Window

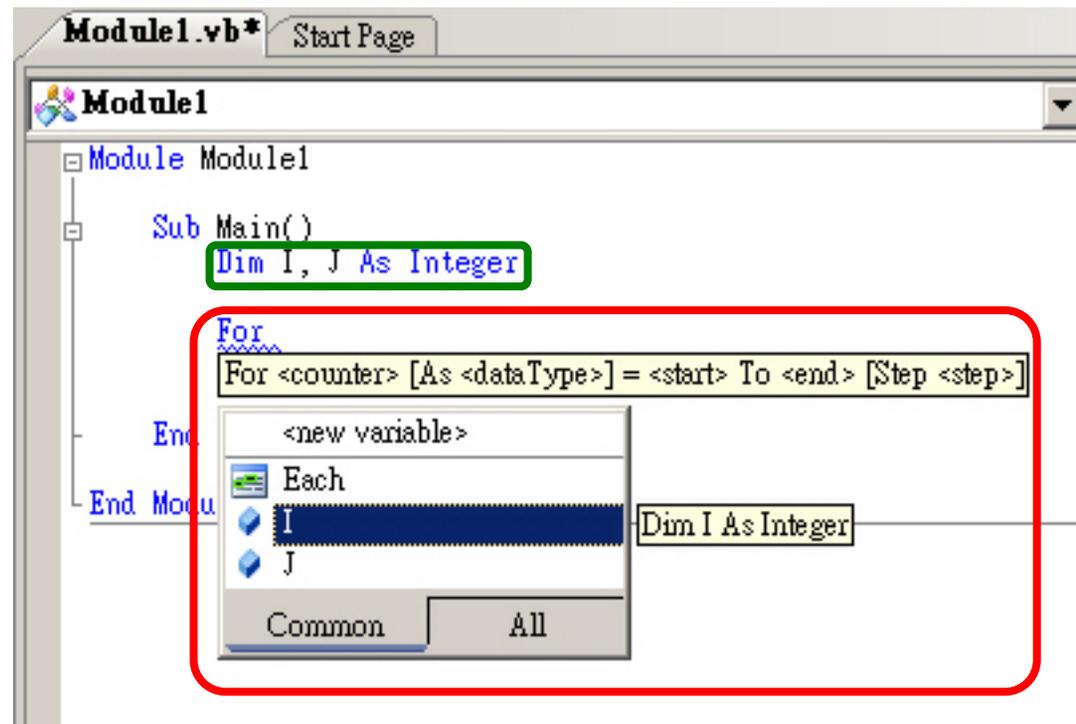
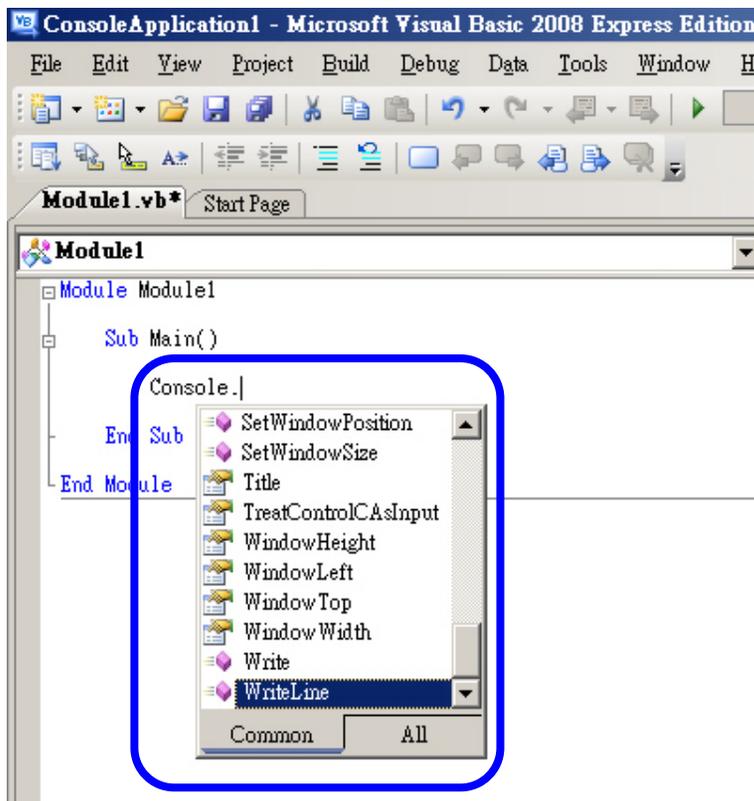
```
alertmsg = "Wrong input!"
? alertmsg.Substring(8)
"put!" {String}
    String: "put!"
? alertmsg.Substring(6, 2)
"in" {String}
    String: "in"
```

## Immediate Window

```
Randomize
? Int (Rnd()*6)+1
2.0
? Int (Rnd()*6)+1
1.0
? Int (Rnd()*6)+1
5.0
? Int (Rnd()*6)+1
4.0
? Int (Rnd()*6)+1
3.0
? Int (Rnd()*6)+1
3.0
? Int (Rnd()*6)+1
4.0
? Int (Rnd()*6)+1
2.0
? Int (Rnd()*6)+1
5.0
? Int (Rnd()*6)+1
1.0
? Int (Rnd()*6)+1
5.0
? Int (Rnd()*6)+1
6.0
```

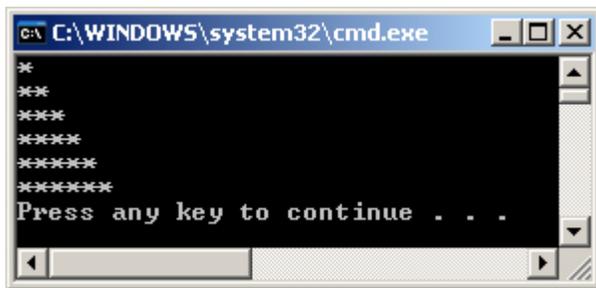
# Minimalist approach

- VB's IntelliSense and code snippets
  - VB's IntelliSense feature offers assistance by displaying possible code snippets
  - Less syntax and typing errors

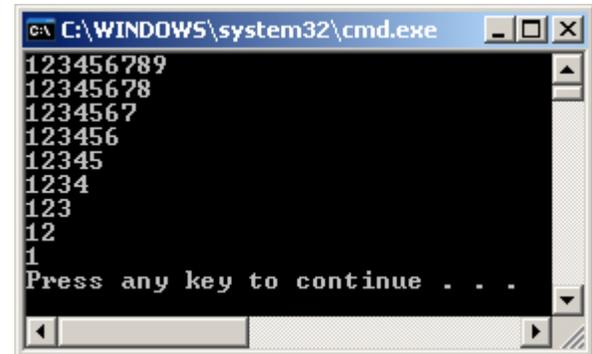


# Minimalist approach

- Students start as soon as possible on meaningful tasks
  - Short and simple programs
  - Programs will be re-used for enhancement
- Examples are followed by similar practical tasks
- Example
- Practical Task



```
C:\WINDOWS\system32\cmd.exe
*
**
***
****
*****
*****
*****
Press any key to continue . . .
```



```
C:\WINDOWS\system32\cmd.exe
123456789
12345678
1234567
123456
12345
1234
123
12
1
Press any key to continue . . .
```

# Minimalist approach

- Examples are re-used and enhanced
  - Students are already familiar with the context and source code
  - Learn by variation

```
1 'This program converts temperature in Fahrenheit to Celsius
2 'and output the Celsius degree to 1 decimal place.
3
4 Module FtoC
5
6 Sub Main()
7     Dim Fahrenheit, Celsius As Single
8
9     'Prompt the user to enter temperature in Fahrenheit
10    Console.WriteLine("Enter Temperature in Fahrenheit: ")
11    Fahrenheit = Console.ReadLine()
12
13    'Convert the temperature in
14    Celsius = (Fahrenheit - 32)
15
16    'Output the temperature in
17    Console.WriteLine("Temperat
18 End Sub
19
20 End Module
```

```
6 Module FtoC
7
8 Sub Main()
9     Dim Fahrenheit, Celsius As Single
10
11    'Prompt the user to enter temperature in Fahrenheit
12    Console.WriteLine("Enter Temperature in Fahrenheit: ")
13    Fahrenheit = Console.ReadLine()
14
15    'Convert the temperature in Fahrenheit to Celsius
16    Celsius = (Fahrenheit - 32) * 5 / 9
17
18    'Output the temperature in Celsius to 1 decimal place
19    Console.WriteLine("Temperature in Celsius is " & "{0:F1}", Celsius)
20
21    'Check a condition whether the Celsius temperature is
22    'greater than or equal to 36.8 and then output the message
23    '!"You have fever." accordingly
24    If Celsius >= 36.8 Then
25        Console.WriteLine("You have fever.")
26    End If
27 End Sub
28
29 End Module
```

# Dissection method

- Dissection is a pedagogical method to
  - highlight key features of the code
  - make a structured walk-through of the code
- Purposes of dissecting a program
  - explain newly encountered programming elements and technical terms in the code
  - explain the code in an easy-to-follow and step-by-step manner
- Each chapter is mostly led by dissecting examples

# Dissection method

```
5
6 Module FtoC
7
8 Sub Main()
9     Dim Fahrenheit, Celsius As Single
10
11     'Prompt the user to enter temperature in Fahrenheit
12     Console.WriteLine("Enter Temperature in Fahrenheit: ")
13     Fahrenheit = Console.ReadLine()
14
15     'Convert the temperature in Fahrenheit to Celsius
16     Celsius = (Fahrenheit - 32) * 5 / 9
17
18     'Output the temperature in Celsius to 1 decimal place
19     Console.WriteLine("Temperature in Celsius is {0:F1} degrees Celsius", Celsius)
20
21     'Check a condition whether the Celsius
22     'greater than or equal to 36.8 and t
23     '"You have fever." accordingly.
24     If Celsius >= 36.8 Then
25         Console.WriteLine("You have fever")
26     End If
27 End Sub
28
29 End Module
```

Two sample outputs are as follows:

A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window shows the program's execution with the following text: "Enter Temperature in Fahrenheit: 98.3", "Temperature in Celsius is 36.8", "You have fever.", and "Press any key to continue . . .". The cursor is positioned at the end of the last line.

Fig. 1.2(a) The condition is met and a message "You have fever." is displayed.

A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window shows the program's execution with the following text: "Enter Temperature in Fahrenheit: 98", "Temperature in Celsius is 36.7", and "Press any key to continue . . .". The cursor is positioned at the end of the last line.

Fig. 1.2(b) The condition is *not* satisfied and the program stops after displaying the temperature in Celsius.

# Dissection method

Lines 24 to 26 are the pieces of code added, apart from additional comments. The statement `If Celsius >= 36.8 Then` in line 24 is a statement to test whether the temperature in Celsius (stored in the variable `Celsius`) is greater than or equal to 36.8. The expression `(Celsius >= 36.8)` in the `If ... Then` statement is called a condition, which consists of comparing the value of a variable using a relational operator, `>=`, with a testing value 36.8. If the condition is met (or satisfied), i.e., the condition is evaluated as true, then the body in line 25 (`Console.WriteLine("You have fever.")`) in the `If ... Then` statement will be executed (**Fig. 1.2(a)**). The body is enclosed with the `If ... Then` statement and the `End If` statement, and `If` and `Then` are keywords of Visual Basic.

# Good programming practice

## Input and Output

VB  
Ch A2

shown below:



```
C:\WINDOWS\system32\cmd.exe
85
Press any key to continue . . . -
```

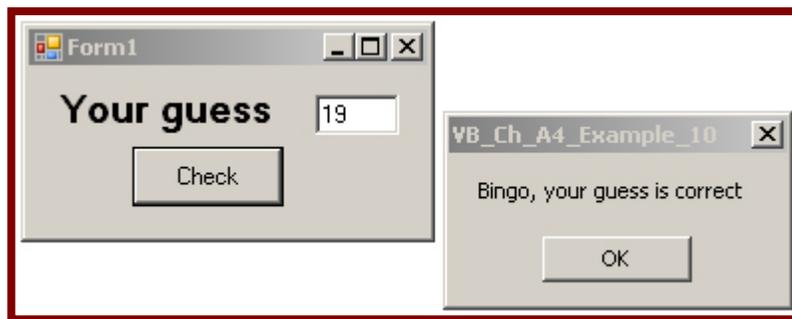
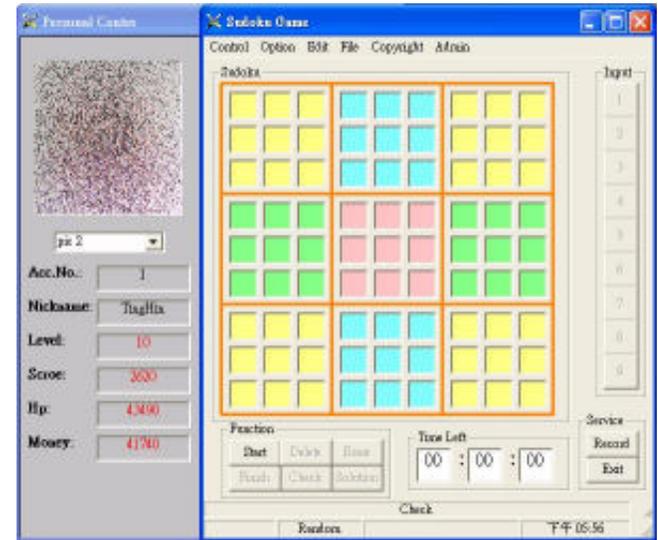
### Good Programming Practice 2.6



*Though numeric variables are initialised to zero by Visual Basic, to avoid confusion and to increase program readability, it is a good habit to initialise variables explicitly.*

# Windows Form Application

- Facilitate students to write Windows application for SBA project if they wish
- As **appendix** and it is **optional**



```
1 Public Class Form1
2     Public HiddenNum As Integer
3
4     Private Sub Form1_Load(ByVal sender As System.Object)
5         Randomize()
6         HiddenNum = Int(Rnd() * 100 + 1)
7
8     End Sub
9
10    Private Sub Button1_Click(ByVal sender As System.Object)
11        Dim Guess As Integer
12
13        Guess = Val(TextBox1.Text)
14        If Guess < HiddenNum Then
15            MsgBox("Your guess is too small")
16        Else
17            If Guess > HiddenNum Then
18                MsgBox("Your guess is too large")
19            Else
20                MsgBox("Bingo, your guess is correct")
21            End If
22        End If
23    End Sub
24 End Class
```

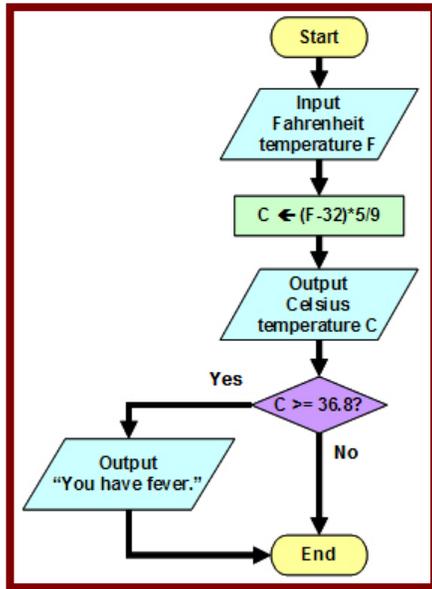
# Overview

- Chap A1 – Getting Familiar with the Programming Environment
- Introduce Object-oriented concept
  - `Console.WriteLine("Hello world")`  

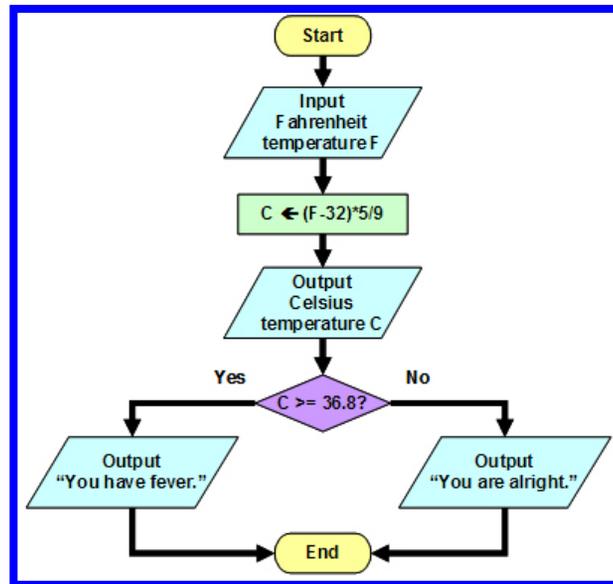
- Chap A2 – Input and Output
  - Input by prompting
- Chap A3 – Arithmetic operations and functions
  - Built-in functions (use of Immediate Windows)

# Overview

- Chap A4 – Selection Structure

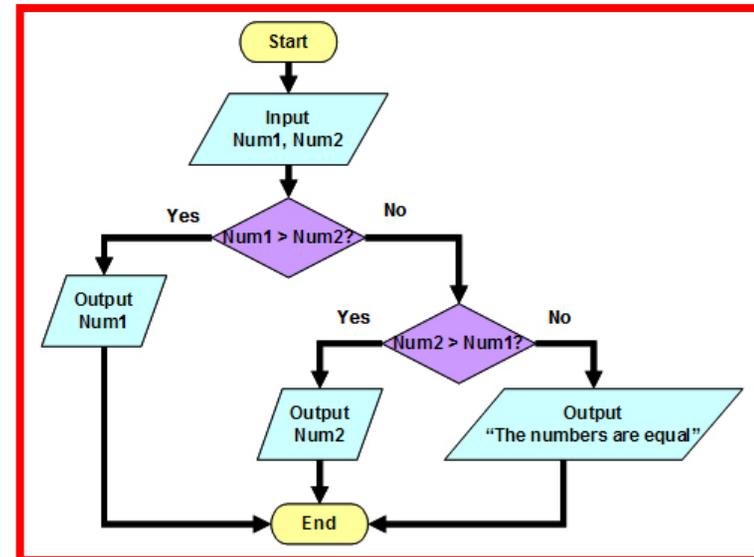


**Single selection**



**Double selection**

## Nested selection



# Overview

- Chap A5 – Iteration Structure
  - Counter, Accumulator, Flag
  - Data validation
  - Terminating value of the inner loop is controlled by the counter variable of the outer loop

# Overview

- Chap A6 – Modular programming
  - Top-down approach, procedures
  - Local variables, global variables
  - Pass-by-value, pass-by-reference
  - User-defined functions
  - Step-wise refinement
  - Stub
  - Interpret errors
  - Debugging
  - Documentation

# Overview

- “Recursion” is introduced in Chap A6 as an alternative solution to the same problem: evaluating factorial
- And it is adopted in merge sort in **Chap A10**

# Overview

- Chap A7 – Arrays
  - Common array manipulations
  - Common pitfall: lower bound of arrays in Visual Basic starts from 0
  - User-defined data type: structure (to be used in [Chap A11](#))
- Chap A8 – Strings
  - Use VB's Immediate Windows to try out many different string functions
- Chap A9 – File Handling
  - Fixed field size (use string functions in [Chap A8](#))
- Chap A10 – Searching, Sorting and Merging
  - Linear search, Binary search
  - Swapping
  - Arrange 2 items in ascending/descending order
  - Bubble sort, Insertion sort, Merge sort
  - Emphasis on algorithm
  - **Aware** the existence of other efficient sorting algorithm, such as quick sort
  - Big O notation

# Overview

## Swapping

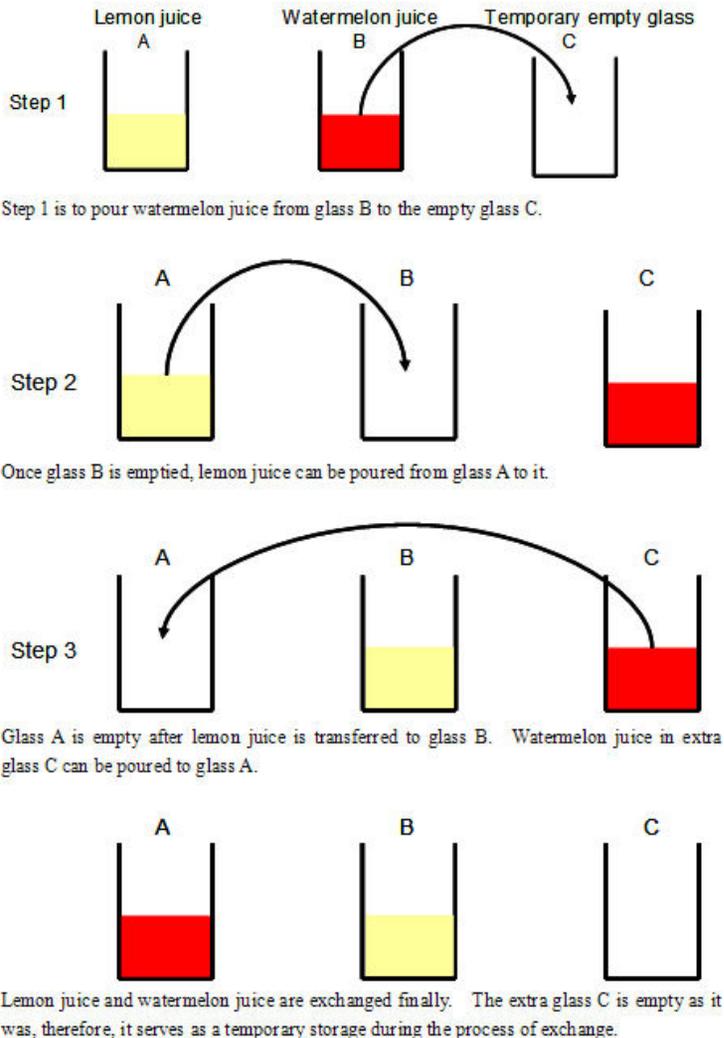
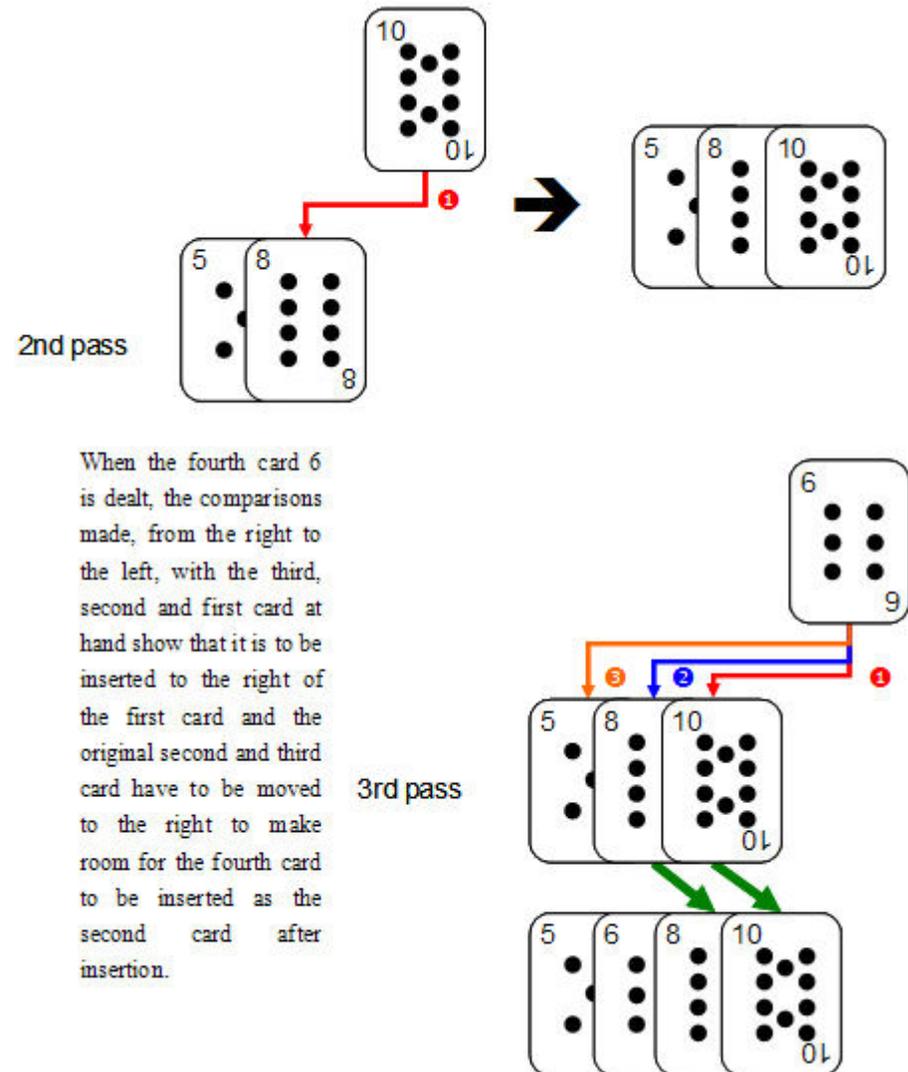


Fig. 2.2 Exchange of juices requires an extra empty glass to hold a kind of juice temporarily.

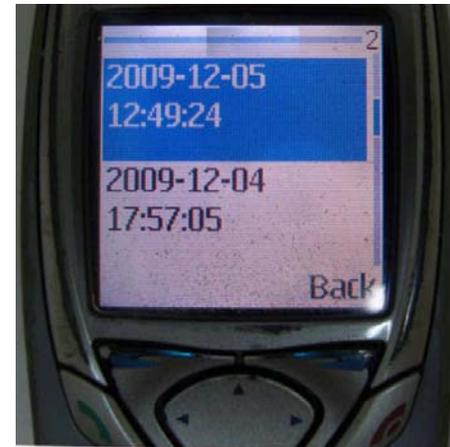
## Insertion sort



# Overview

- Chap A11 – Stacks, Queues and Linked Lists
  - Implement in terms of arrays
  - No library function call

```
30 Sub Push(ByVal towel_colour As String, ByRef towel As stack_towel)
31     'upper bound of an array of a stack is stack_size - 1 because
32     'lower bound of an array in VB starts from 0
33     If towel.stack_pointer = stack_size - 1 Then
34         Console.WriteLine("The stack is full.")
35     Else
36         towel.stack_pointer += 1
37         towel.element(towel.stack_pointer) = towel_colour
38     End If
39 End Sub
```



Application of a Stack  
– retrieve most recent items  
(most recent calls from  
Call register of a mobile phone)

- Linear queue and its drawbacks → Circular queue

# Overview

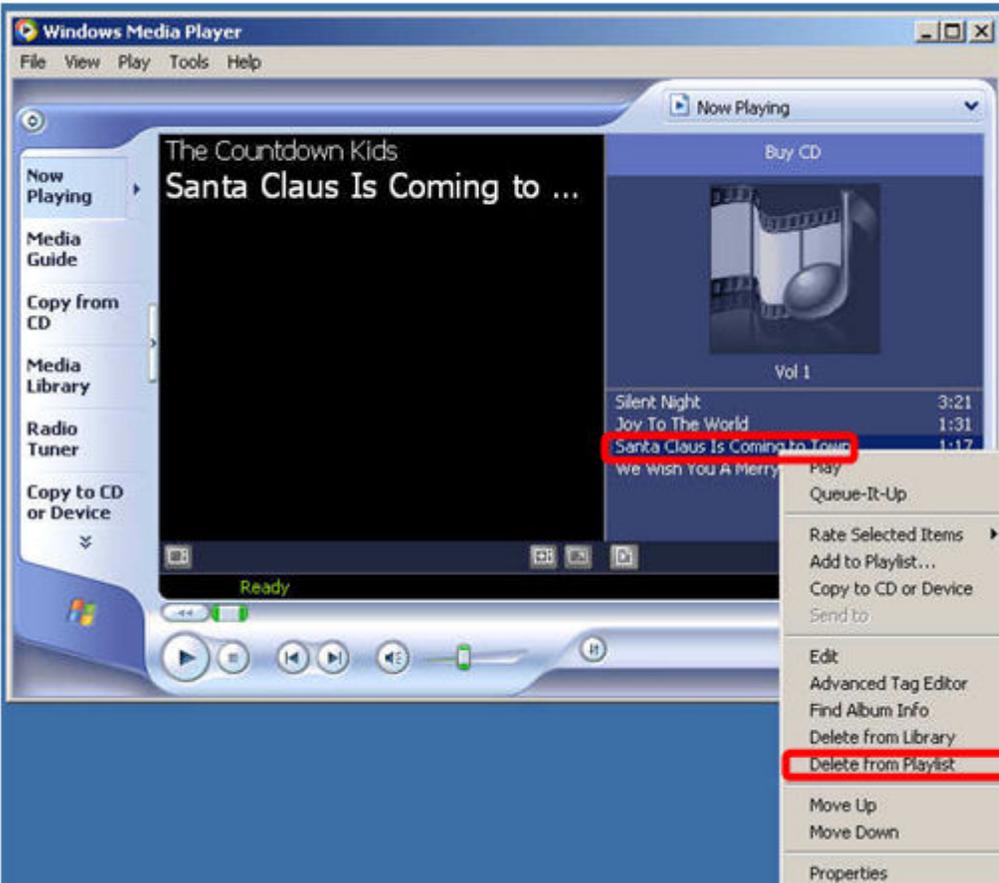
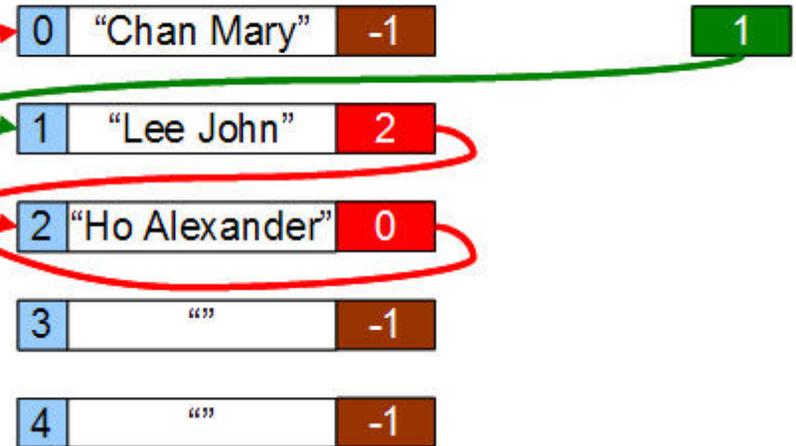
- Linked List
  - Next pointer, array index

index in the array

item

next pointer

head pointer



# Learning Task

- Consolidate students' learning after they have learnt a few chapters
- Go through a series of 4 learning tasks to complete a mini-project by implementing different features gradually
- The 4 tasks are NOT fragmented, but inter-related so that they can be combined to become a computer game
- Students gain experience to accomplish a project similar to a SBA project

# Learning Task

## Mastermind game



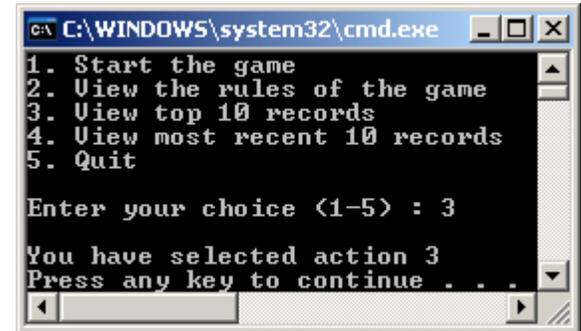
- Player tries to decode the secret code of 4 colour code pegs
- Codemaker gives feedback to the player for pegs of correct colour and in the correct position, and pegs of correct colour but in wrong position

# Learning Task A #1

- Chap A1 – A3
  - Getting familiar with the programming environment
  - Input and Output
  - Arithmetic Operations and Functions

# Learning Task A #1

- Design a menu (Chap A2)
  - Input by prompting



```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> : 3

You have selected action 3
Press any key to continue . . .
```

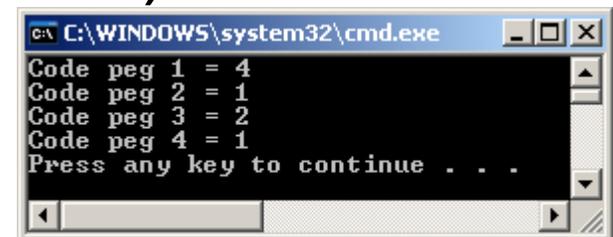
- Display in colour (Chap A1)
  - Learn by doing something unfamiliar, e.g.



```
C:\WINDOWS\system32\cmd.exe
RCYMB
Press any key to continue . . .
```

- `Console.ForegroundColor = ConsoleColor.Red`

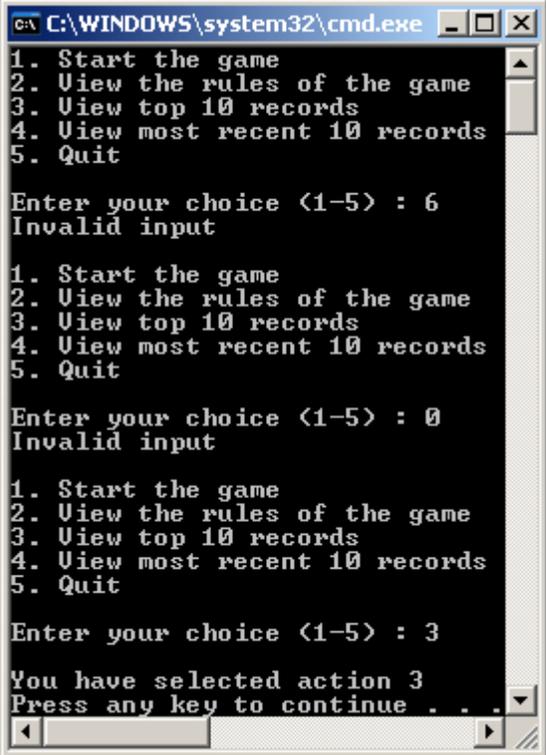
- Generate secret code (Chap A3)
  - Generate random numbers



```
C:\WINDOWS\system32\cmd.exe
Code peg 1 = 4
Code peg 2 = 1
Code peg 3 = 2
Code peg 4 = 1
Press any key to continue . . .
```

# Learning Task A #2

- Validate player's choice (Chap A4, A5)
  - Flag
  - Selection Structure
  - Iteration Structure



```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> : 6
Invalid input

1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> : 0
Invalid input

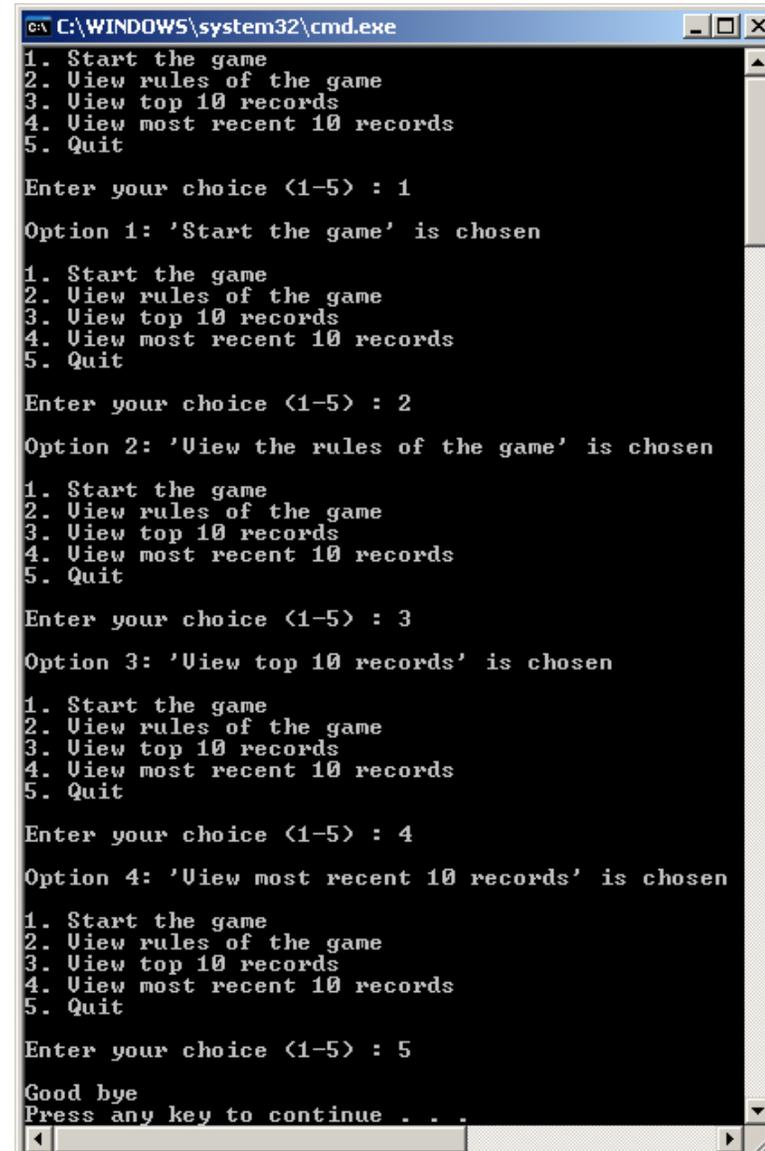
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> : 3

You have selected action 3
Press any key to continue . . .
```

# Learning Task A #2

- Act according to player's choice (Chap A6)
  - Procedure (sub-routines)
  - Stub programming



```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit
Enter your choice <1-5> : 1
Option 1: 'Start the game' is chosen
1. Start the game
2. View rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit
Enter your choice <1-5> : 2
Option 2: 'View the rules of the game' is chosen
1. Start the game
2. View rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit
Enter your choice <1-5> : 3
Option 3: 'View top 10 records' is chosen
1. Start the game
2. View rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit
Enter your choice <1-5> : 4
Option 4: 'View most recent 10 records' is chosen
1. Start the game
2. View rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit
Enter your choice <1-5> : 5
Good bye
Press any key to continue . . .
```

# Learning Task A #2

- Play a dummy game (Chap A4, A5, A6)
  - Counting
  - Selection Structure
  - Iteration Structure
  - Testing & Debugging

```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> : 1

Trial 1
Guess 1 : 1
Guess 2 : 2
Guess 3 : 3
Guess 4 : 4
Trial 2
Guess 1 : 4
Guess 2 : 3
Guess 3 : 2
Guess 4 : 1
Trial 3
Guess 1 : 4
Guess 2 : 1
Guess 3 : 2
Guess 4 : 1

Great, you've guessed in 3 trials.

1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> :
```

```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> : 1

Trial 1
Guess 1 : 1
Guess 2 : 2
Guess 3 : 3
Guess 4 : 4
Trial 2
Guess 1 : 4
Guess 2 : 3
Guess 3 : 2
Guess 4 : 1
Trial 3
Guess 1 : 1
Guess 2 : 1
Guess 3 : 1
Guess 4 : 1
Trial 4
Guess 1 : 4
Guess 2 : 4
Guess 3 : 4
Guess 4 : 4
Trial 5
Guess 1 : 3
Guess 2 : 3
Guess 3 : 3
Guess 4 : 3

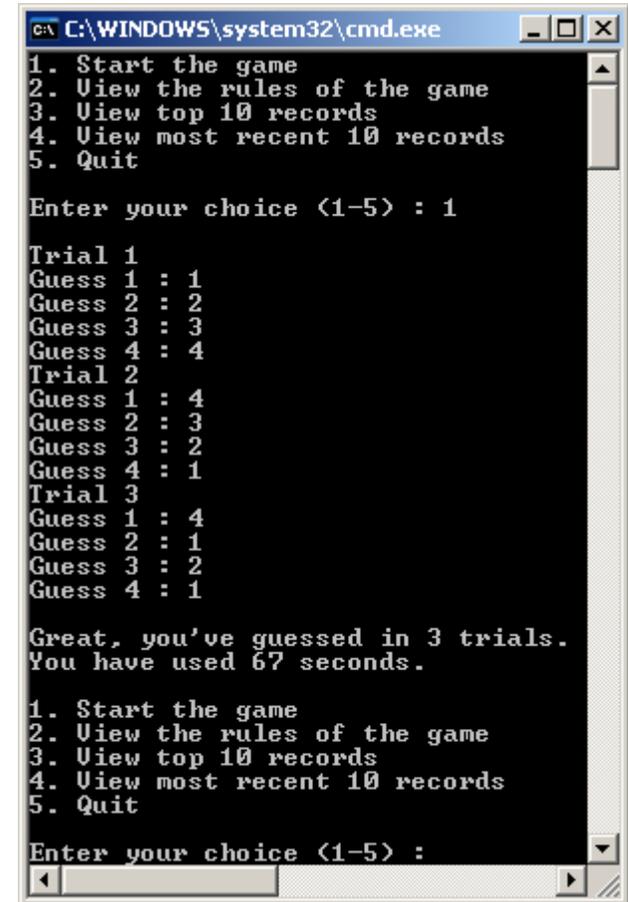
The secret code is 4 1 2 1

1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> :
```

# Learning Task A #2

- Time the game
  - Learn by doing something unfamiliar, e.g. get the system time and store as a date data type



```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> : 1

Trial 1
Guess 1 : 1
Guess 2 : 2
Guess 3 : 3
Guess 4 : 4
Trial 2
Guess 1 : 4
Guess 2 : 3
Guess 3 : 2
Guess 4 : 1
Trial 3
Guess 1 : 4
Guess 2 : 1
Guess 3 : 2
Guess 4 : 1

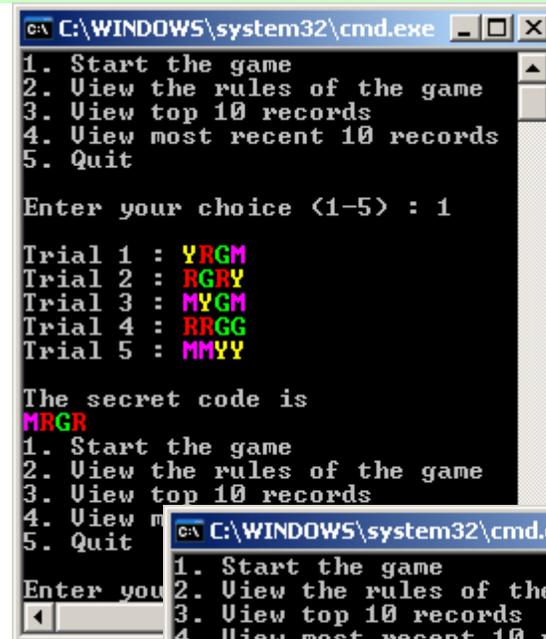
Great, you've guessed in 3 trials.
You have used 67 seconds.

1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

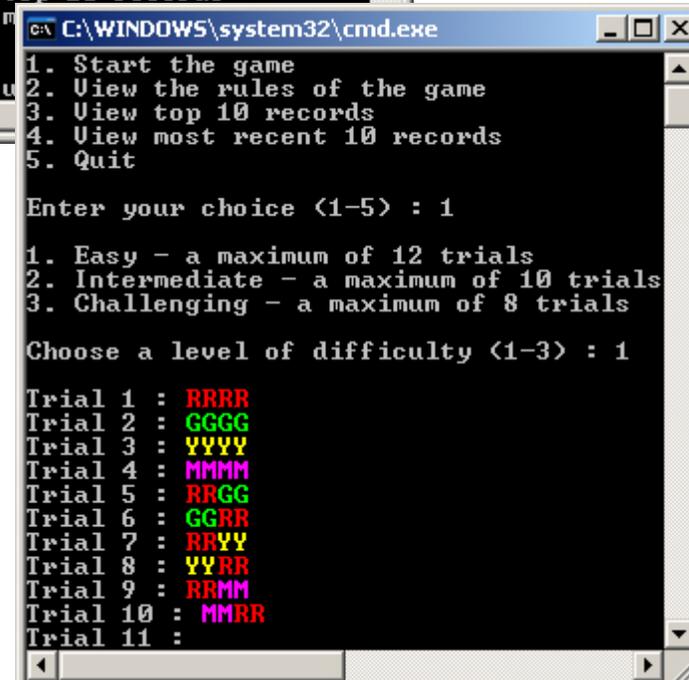
Enter your choice <1-5> :
```

# Learning Task A #3

- Improve the data structure and user-interface (Chap A7, A8)
  - Array
  - Read single character from a keystroke
- Implement level of difficulty



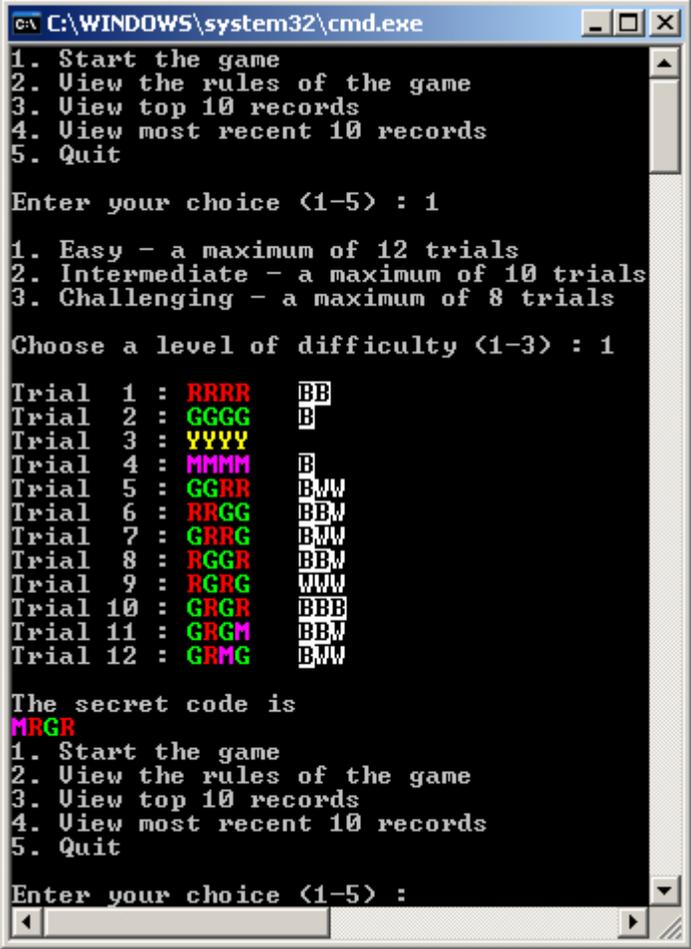
```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit
Enter your choice <1-5> : 1
Trial 1 : YRGM
Trial 2 : RGRY
Trial 3 : MYGM
Trial 4 : RRGG
Trial 5 : MMYM
The secret code is
MRGR
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit
Enter your choice <1-5> :
```



```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit
Enter your choice <1-5> : 1
1. Easy - a maximum of 12 trials
2. Intermediate - a maximum of 10 trials
3. Challenging - a maximum of 8 trials
Choose a level of difficulty <1-3> : 1
Trial 1 : RRRR
Trial 2 : GGGG
Trial 3 : YYYM
Trial 4 : MMMM
Trial 5 : RRGG
Trial 6 : GGRR
Trial 7 : RRYM
Trial 8 : YYRR
Trial 9 : RRMM
Trial 10 : MMRR
Trial 11 :
```

# Learning Task A #3

- Check the player's guess
  - Algorithm
    - Count number of pegs of correct colour in the correct position (Black) first
    - Count number of pegs of correct colour but in wrong position



```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> : 1

1. Easy - a maximum of 12 trials
2. Intermediate - a maximum of 10 trials
3. Challenging - a maximum of 8 trials

Choose a level of difficulty <1-3> : 1

Trial 1 : RRRR BB
Trial 2 : GGGG B
Trial 3 : YYY Y
Trial 4 : MMM B
Trial 5 : GRR BWW
Trial 6 : RRG BBW
Trial 7 : RRR BWW
Trial 8 : RGR BBW
Trial 9 : RGR WWW
Trial 10 : GRG BBB
Trial 11 : GRM BBW
Trial 12 : GRM BWW

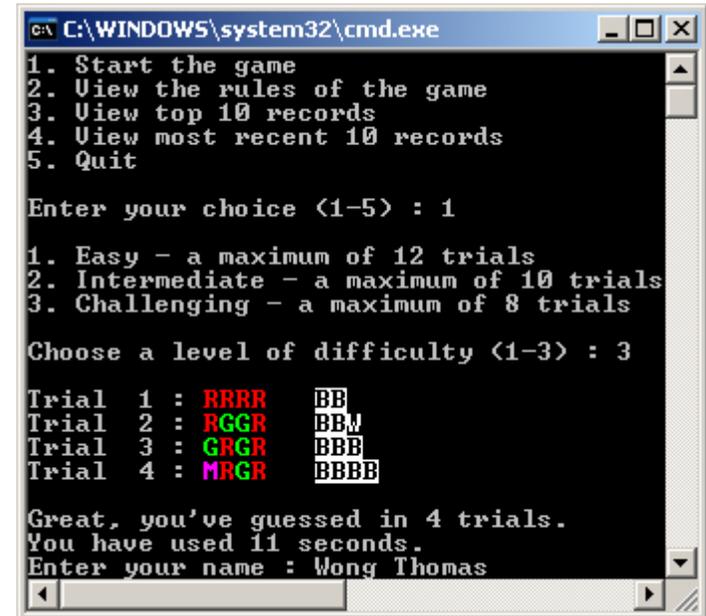
The secret code is
MRGR

1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> :
```

# Learning Task A #3

- Save winner's details to file (Chap A8, A9)
  - Strings
  - File Handling



```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

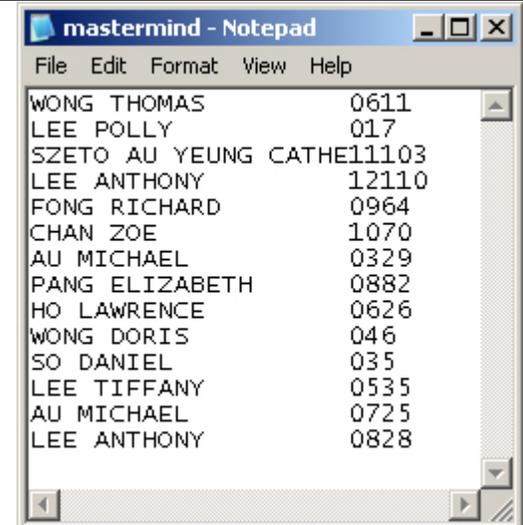
Enter your choice <1-5> : 1

1. Easy - a maximum of 12 trials
2. Intermediate - a maximum of 10 trials
3. Challenging - a maximum of 8 trials

Choose a level of difficulty <1-3> : 3

Trial 1 : RRRR BB
Trial 2 : RGGR BBW
Trial 3 : GRGR BBB
Trial 4 : MRGR BBBB

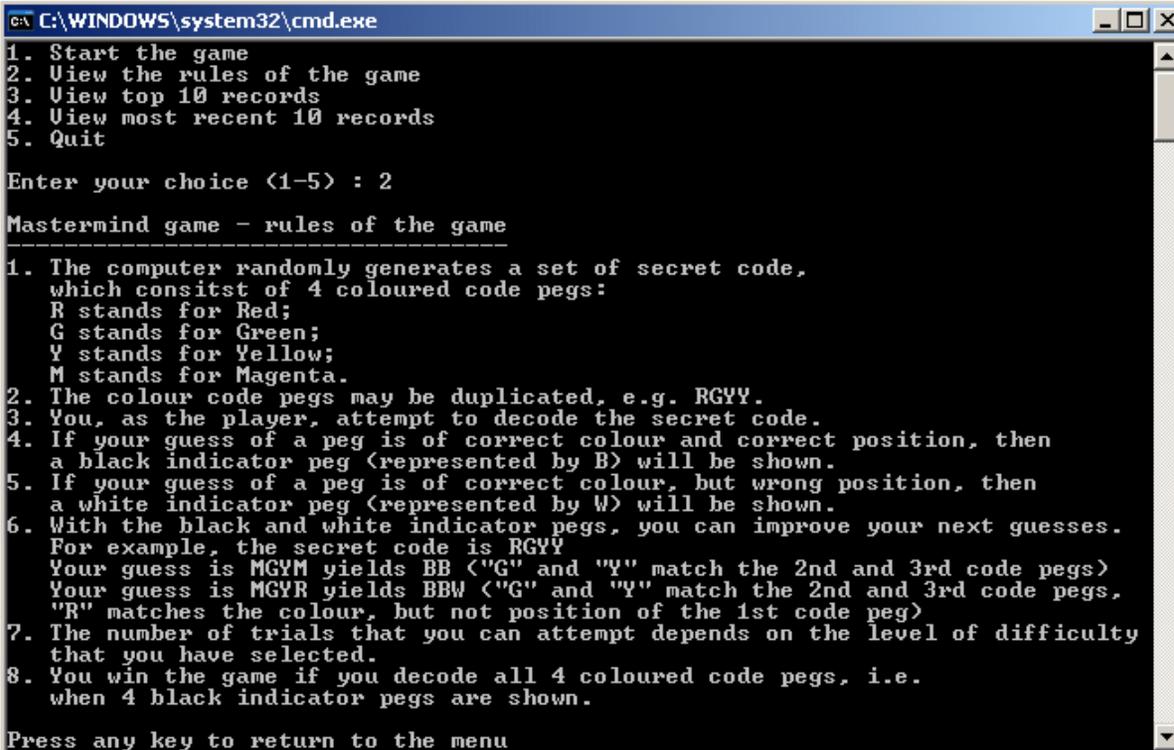
Great, you've guessed in 4 trials.
You have used 11 seconds.
Enter your name : Wong Thomas
```



```
mastermind - Notepad
File Edit Format View Help
WONG THOMAS      0611
LEE POLLY        017
SZETO AU YEUNG CATHE11103
LEE ANTHONY      12110
FONG RICHARD     0964
CHAN ZOE         1070
AU MICHAEL       0329
PANG ELIZABETH  0882
HO LAWRENCE     0626
WONG DORIS       046
SO DANIEL        035
LEE TIFFANY      0535
AU MICHAEL       0725
LEE ANTHONY      0828
```

# Learning Task A #3

- Read the rules of the game from file and display (Chap A9)
  - File Handling



```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

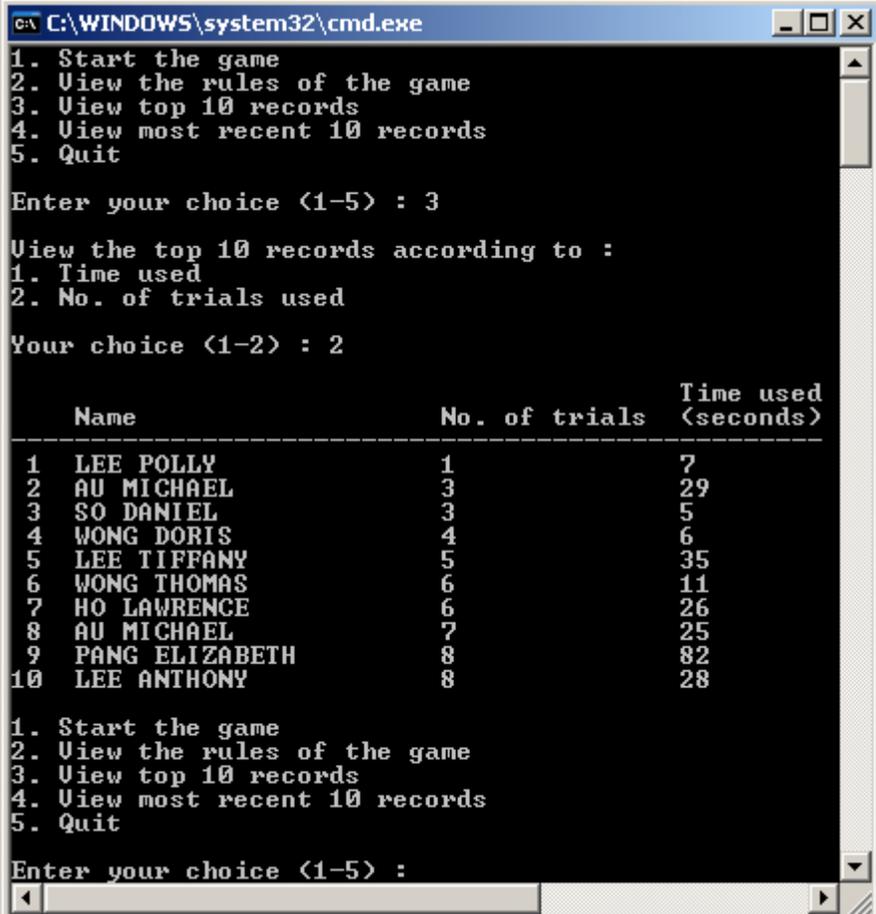
Enter your choice (1-5) : 2

Mastermind game - rules of the game
-----
1. The computer randomly generates a set of secret code,
   which consist of 4 coloured code pegs:
   R stands for Red;
   G stands for Green;
   Y stands for Yellow;
   M stands for Magenta.
2. The colour code pegs may be duplicated, e.g. RGYG.
3. You, as the player, attempt to decode the secret code.
4. If your guess of a peg is of correct colour and correct position, then
   a black indicator peg (represented by B) will be shown.
5. If your guess of a peg is of correct colour, but wrong position, then
   a white indicator peg (represented by W) will be shown.
6. With the black and white indicator pegs, you can improve your next guesses.
   For example, the secret code is RGYG
   Your guess is MGYM yields BB (<"G" and "Y" match the 2nd and 3rd code pegs)
   Your guess is MGYR yields BBW (<"G" and "Y" match the 2nd and 3rd code pegs,
   "R" matches the colour, but not position of the 1st code peg)
7. The number of trials that you can attempt depends on the level of difficulty
   that you have selected.
8. You win the game if you decode all 4 coloured code pegs, i.e.
   when 4 black indicator pegs are shown.

Press any key to return to the menu
```

# Learning Task A #4

- Sort the records of the winners (Chap A10)
  - Sorting



```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> : 3

View the top 10 records according to :
1. Time used
2. No. of trials used

Your choice <1-2> : 2

Name                               No. of trials  Time used
-----
1  LEE POLLY                               1             7
2  AU MICHAEL                              3            29
3  SO DANIEL                                3             5
4  WONG DORIS                               4             6
5  LEE TIFFANY                              5            35
6  WONG THOMAS                              6            11
7  HO LAWRENCE                              6            26
8  AU MICHAEL                               7            25
9  PANG ELIZABETH                           8            82
10 LEE ANTHONY                              8            28

1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> :
```

# Learning Task A #4

- Retrieve most recent records (Chap A11)
  - Stacks

```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> : 4

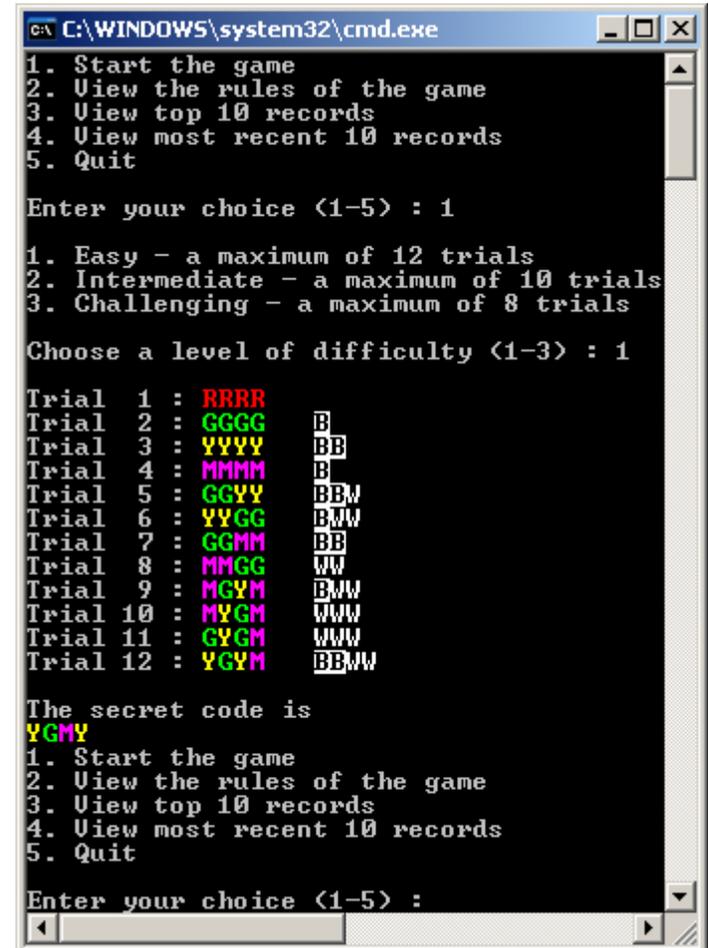
      Name                               No. of trials  Time used
      -----                               -
14  LEE ANTHONY                           8             28
13  AU MICHAEL                             7             25
12  LEE TIFFANY                             5             35
11  SO DANIEL                               3              5
10  WONG DORIS                              4              6
 9  HO LAWRENCE                             6             26
 8  PANG ELIZABETH                         8             82
 7  AU MICHAEL                              3             29
 6  CHAN ZOE                               10            70
 5  FONG RICHARD                            9             64

1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> :
```

# Learning Task A #4

- Finalize the game by removing the pre-set secret code



```
C:\WINDOWS\system32\cmd.exe
1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> : 1

1. Easy - a maximum of 12 trials
2. Intermediate - a maximum of 10 trials
3. Challenging - a maximum of 8 trials

Choose a level of difficulty <1-3> : 1

Trial 1 : RRRR
Trial 2 : GGGG
Trial 3 : YYY Y
Trial 4 : MMMM
Trial 5 : GGY Y
Trial 6 : YYGG
Trial 7 : GGMM
Trial 8 : MMGG
Trial 9 : MGYM
Trial 10 : MYGM
Trial 11 : GYGM
Trial 12 : YGYM

B
BB
B
BBW
BW
BB
W
BW
WWW
WWW
WWW
BBW

The secret code is
YGYM

1. Start the game
2. View the rules of the game
3. View top 10 records
4. View most recent 10 records
5. Quit

Enter your choice <1-5> :
```

# Assessment Task A #1 – #4

- Chapter A1 Getting familiar with the programming environment
- Chapter A2 Input and output
- Chapter A3 Arithmetic operations and functions

## Exercise / Quiz

1. Which of the following is a valid variable name in VB?
- A. MyAge2010
  - B. 2010MyAge
  - C. \_MyAge
  - D. My Age

```

8      Function Function2(ByVal x() As Integer, By
As Integer) As Integer
9          Dim i, smallest_position As Integer
10         smallest_position = start_position
11         For i = start_position To x.GetUpperBo
12             If x(i) <  Then
13                 
14             End If
15         Next
16         Return()
17     End Function
    
```

Line no.	Statement
1	Function Encryption(ByVal x As String) As String
2	Dim i As Integer
3	Dim result As String
4	result = ""
5	For i = 0 To x.Length - 1
6	result = result & Chr(Asc(x.Substring(i, 1)) + 3)
7	Next
8	Return(result)
9	
10	Sub Main
11	Console.WriteLine(Encryption("gold4"))
12	End Sub

(a) Fill in the following table after the statement in line 6 is executed in each pass.

pass	Asc(x.Substring(i, 1))	Chr(SubString(i, 1)) + 3)	result
1			
2			
3			
4			
5			

# Assessment Task A #1 – #4

12. Tic-Tac-Toe is a famous game for 2 players to take turn to mark crosses ("X") and noughts ("O") on spaces of a 3 by 3 grid. A player wins the game if he is the first one to mark 3 of his symbols either in a horizontal, vertical or diagonal row, e.g.

X	X	X

	X	
	X	
	X	

		X
	X	
X		

A one-dimensional array `grid(8)` is declared as `Char` type to represent spaces of the grid in the following way:

0	1	2
3	4	5
6	7	8

For instance, a cross is assigned to the centre space with  
`grid(4) = "X"`

A nought is assigned to the lower right corner with the  
`grid(8) = "O"`

- (c) Write a user-defined function `Player_X_Win_Horizontal` to return a Boolean value `True` if the player who marks crosses wins by marking 3 crosses at one of the 3 horizontal rows. Otherwise, the function returns a Boolean value `False`. The array `grid` is passed to the function as a parameter.

```
Function Player_X_Win_Horizontal(ByVal g() As Char) As Boolean
```

```
End Function
```

# Teaching plan for each chapter

## Teaching plan for topic A: Programming

VB

### Chapter A2 – Input and Output

#### Objectives

1. To understand how to use the methods `WriteLine` and `Write` for output and recognise their difference.
2. To use format specifier to display formatted output.
3. To declare constants and variables with appropriate data type.
4. To use assignment statements to assign values to a constant and a variable.
5. To understand how to use the method `ReadLine` for input.
6. To recognise input by prompting improves user-friendliness of a program.

#### Duration

2 hours in total (approximately equivalent to 3 periods, each period lasts 40 minutes)

#### Resources

1. Learning task sheet A #1 (The learning task should be attempted after completing chapter A3.)
2. Assessment task sheet A #1 (The assessment task should be attempted after completing chapter A3.)

#### Suggested time allocation

Learning outcomes	Duration in minutes	Section /Resources in the Package	Learning & Teaching activities	Teaching points / Explanatory notes
1. Identify the inputs and outputs involved in a solution.	40 min (1 period)	● Section 1	● Introduce the output method <code>Write</code> and recognise the difference between the methods	● Direct instruction. ● Compare the effects of using the methods <code>Write</code> and <code>WriteLine</code> .

# Topic B

Programming Languages

# Topic B: Programming Languages

B1	Programming paradigms	10½ periods	7 hours
B2	Language translators and compilers	7½ periods	5 hours
Total:		<b>18 periods</b>	<b>12 hours</b>

- Briefly in the current HKCEE CIT

## HKCEE CIT

- Selection of programming languages
  - Students should be aware that there exist different types of programming paradigm, such as imperative programming, object-oriented programming, logic programming, etc.
  - Students should also be aware that different programming languages are designed to satisfy different specific purposes.

# Topic B: Programming Languages

- But, it is required in details in the current HKAL CS

---

## *B. Programming Languages*

HKAL CS

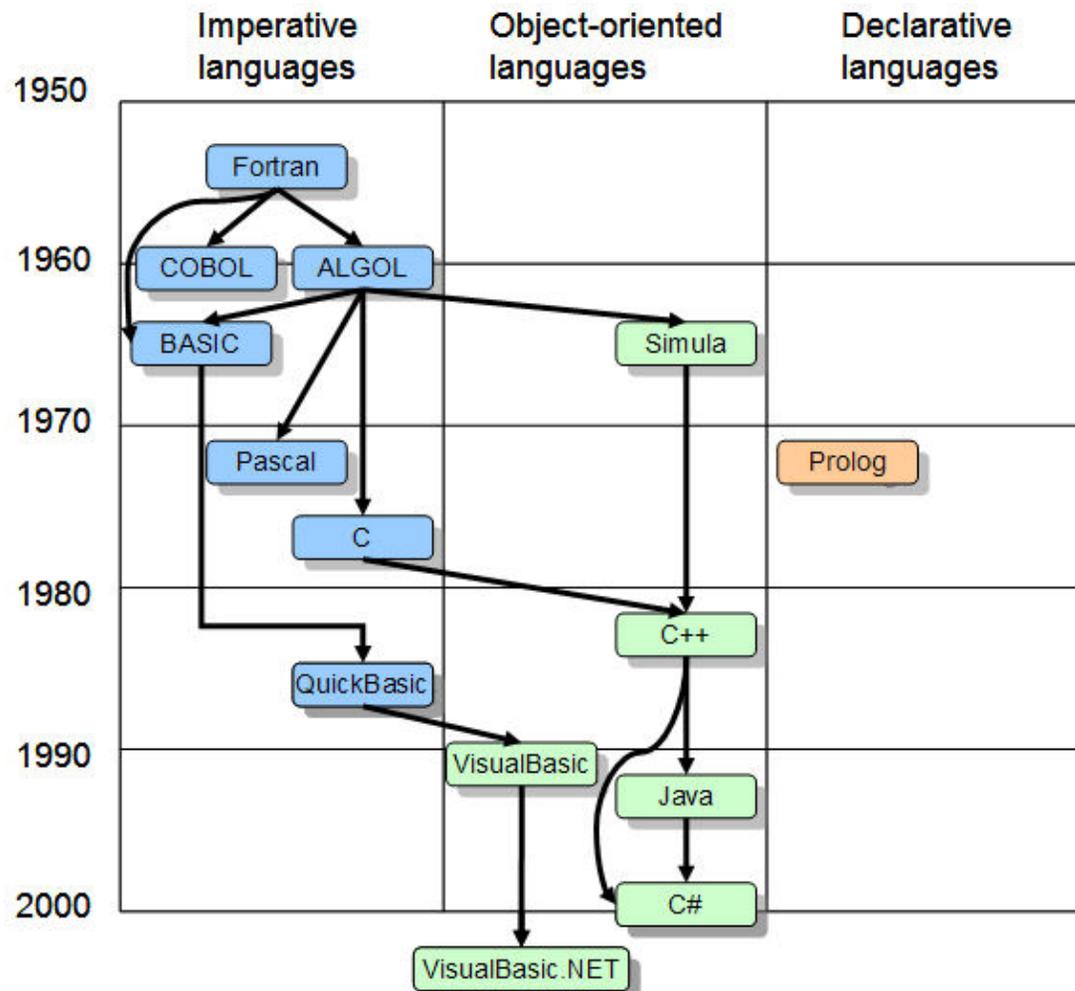
(9 hours)

- Evaluation of programming languages
  - Students should develop a basic understanding of the programming paradigms involved in procedural, declarative, object-oriented, visual and query languages and be able to describe the criteria for selecting a programming language for problem-solving in a given context.
- Program translation and stages of compilation
  - Students should have a basic understanding of the need for high-level programming languages and the processes involved in the translation and analysis of such programs. Concepts include lexical, syntactic and semantic analysis, code generation, linkers and loaders, and compilers and interpreters.

- 
- **Topic B is pitched to HKAL CS standard**

# Evolution of programming languages

- Low-level (Machine code, Assembly language) vs. High-level programming languages (3GL, 4GL, 5GL)



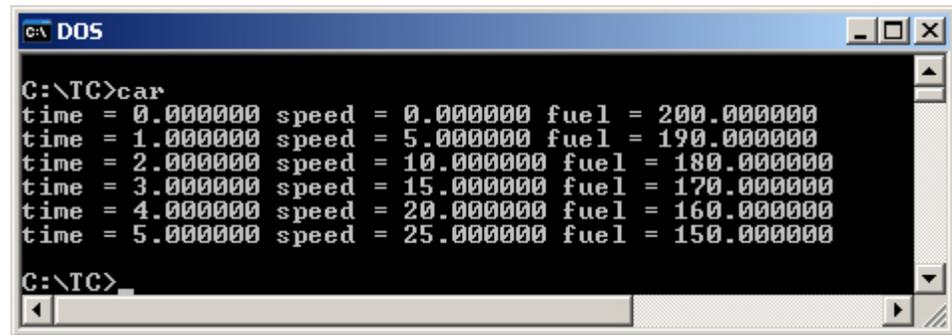
# Programming paradigms

- Programming paradigms
  - Procedural (imperative)
  - Object-oriented
  - Declarative
  - Query (designed for DBMS; not generic and not a paradigm)

<b>b. Programming Languages (12 hours)</b>		
<b>i. Programming paradigms</b>	<ul style="list-style-type: none"><li>• Be aware of the evolution of programming languages.</li><li>• Recognise the programming paradigms involved in procedural, logic, object-oriented and query languages.</li><li>• Describe the criteria for selecting a programming language for a specific problem.</li></ul>	One programming language for each paradigm is selected for illustration.

# Procedural programming paradigm

- Key concepts
  - Variables
  - Commands
  - Procedures

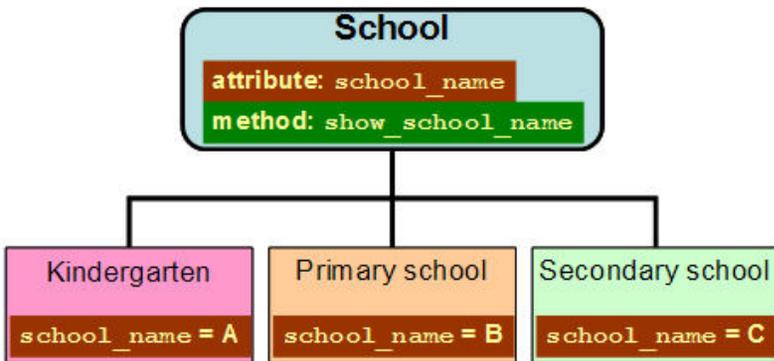
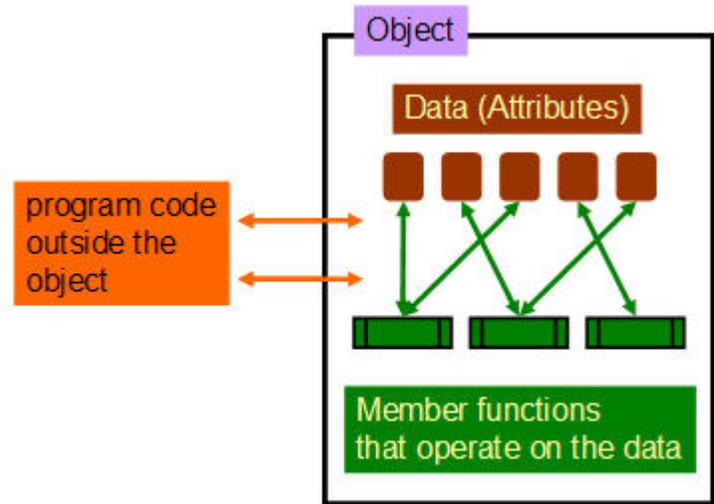


```
C:\DOS
C:\TC>car
time = 0.000000 speed = 0.000000 fuel = 200.000000
time = 1.000000 speed = 5.000000 fuel = 190.000000
time = 2.000000 speed = 10.000000 fuel = 180.000000
time = 3.000000 speed = 15.000000 fuel = 170.000000
time = 4.000000 speed = 20.000000 fuel = 160.000000
time = 5.000000 speed = 25.000000 fuel = 150.000000
C:\TC>
```

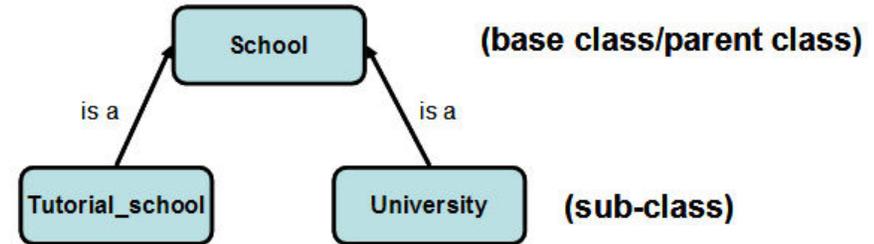
- Procedure is the representation of the method of solving a real-world problem
- Advantages and disadvantages

# Object-oriented programming paradigm

- Object and classes
- Encapsulation
- Inheritance
- Polymorphism



(Class)



(Objects)

# Object-oriented programming paradigm

- Object and classes
- Encapsulation
- Inheritance
- Polymorphism

```
Public Class School
    Private school_name As String
    Private school_address As String

    Public Sub New()
        'Program statements to be coded to construct a new school
    End Sub

    Private Sub show_school_name()
        'Program statements to be coded to display the variable school_name
    End Sub
End Class
```

# Declarative programming paradigm

- Students should practice so as to experience the difference between declarative programming paradigm and procedural/OO programming paradigms
- Recap 5GL is focused on “What to solve” rather than “How to solve”
- Free Prolog compilers available, e.g.
  - GNU Prolog
  - SWI Prolog

# Prolog

- Easy to start the hands-on practice
  - Notepad Editor / Built-in Editor
  - Enter “fact” and “rule”
  - Make Query

```
family - Notepad
File Edit Format View Help
/* A Prolog program to trace along a family tree */
/* facts */
male(alan).
male(billy).
male(clive).
male(david).
parent(alan, billy).
parent(alan, clive).
/* rules */
father(F,X):- parent(F,X), male(F).
```

```
SWI-Prolog (Multi-threaded, version 5.6.64)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 5.6.64)
Copyright (c) 1990-2008 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?-
% c:/family.pl compiled 0.00 sec, 1,492 bytes
1 ?- male(billy).
true.

2 ?- parent(X, clive).
X = alan.

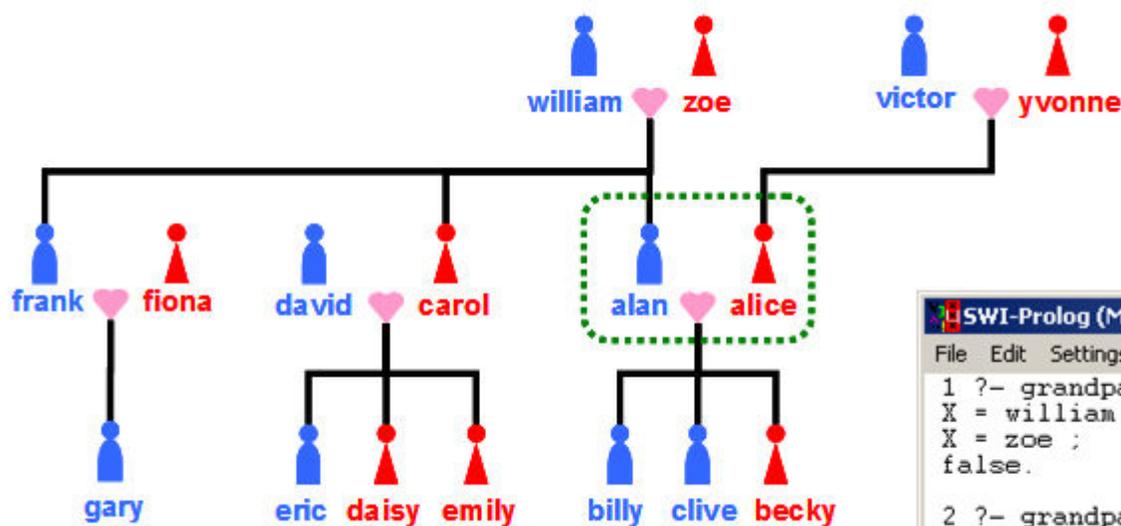
3 ?- father(alan, Y).
Y = billy ;
Y = clive.

4 ?- parent(Z, david).
false.

5 ?-
```

# Prolog

- Learning Task B #1



- male, female
- father, mother, parent
- sibling
- grandparent, cousin

```
SWI-Prolog (Multi-threaded, version 5.6.64)
File Edit Settings Run Debug Help
1 ?- grandparent(X, gary).
X = william ;
X = zoe ;
false.

2 ?- grandparent(william, X).
X = billy ;
X = clive ;
X = becky ;
X = gary ;
X = eric ;
X = daisy ;
X = emily.

3 ?- cousin(X, becky).
X = gary ;
X = eric ;
X = daisy ;
X = emily ;
false.

4 ?- █
```

# Query language

- Experience query language by practicing



Employee_ID	Name	Sex	DOB	Department
07001	Wong Tai Man	M	1985.01.13	Administration
07002	Lee Siu Mei	F	1984.11.25	Accounting
07003	Lau Bo Bo	F	1983.07.31	Sales
08002	Wong Lok Tin	M	1987.04.09	Sales
08003	Lee Chi fai	M	1986.12.10	Accounting
09001	Chan Amy	F	1983.09.17	Sales



```
SELECT Name, Department
FROM Employee
WHERE Sex = 'F'
ORDER BY Name;
```



Name	Department
Chan Amy	Sales
Lau Bo Bo	Sales
Lee Siu Mei	Accounting

# Criteria of selecting a programming language

- No universal rules, but some criteria are provided for consideration
- **Justify** decision after comparing several candidate programming languages
- Case study: Learning Task B #2, Assessment Task B #1

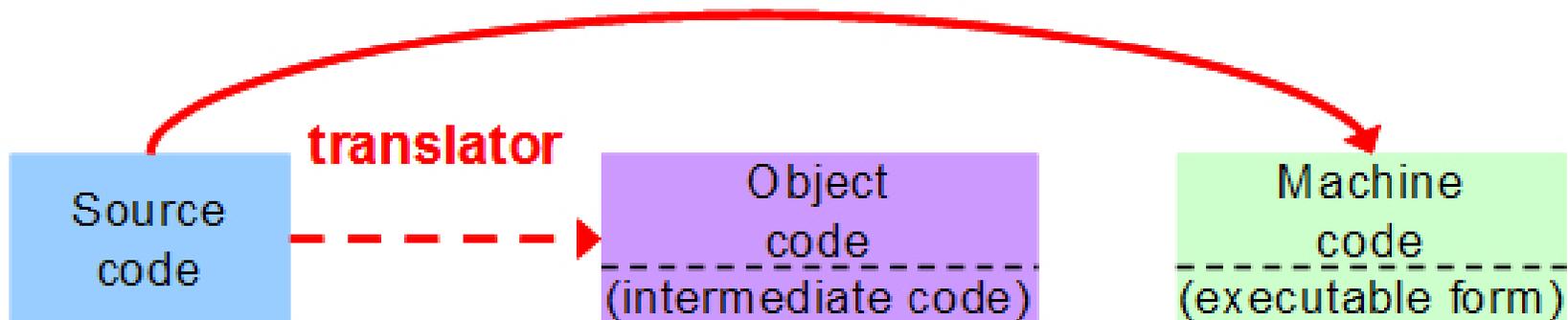
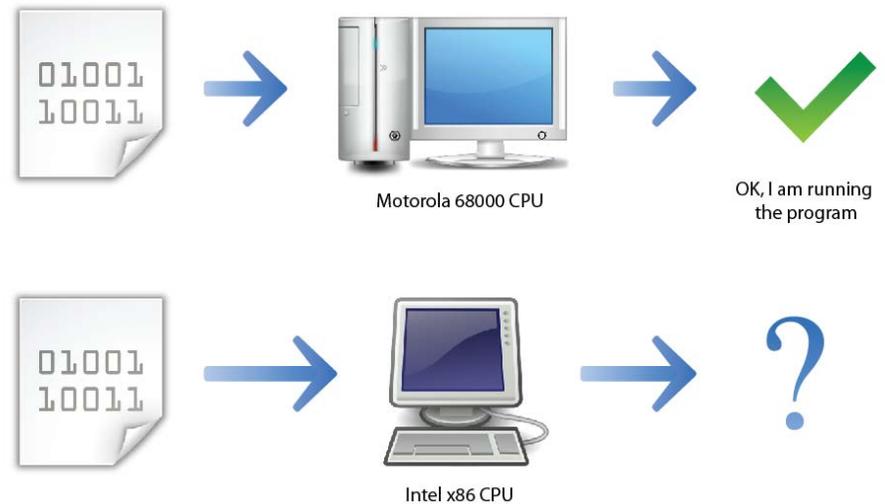
# Criteria of selecting a programming language

- Learning task: Develop a small-scale library system
- Assessment task: Develop a program to be used in sports day

Criterion	Description	Relevant requirement code	Findings	Pascal	C	Visual Basic	Java
Execution efficiency	It refers to the efficient execution of non-I/O computational operations.	B2	C language compiler can generate efficient executable code.		✓		
Portability	It is used in the narrow sense that the source code can be easily compiled to executable codes on different operating systems.	C1	C language and Java have very good portability across a wide range of computer platforms.		✓		✓
Readability	It means whether a program written in a language can be read and understood with ease.	C2	As long as a programmer has a good programming habit, the code should be easily understood by others, no matter which programming language he uses to code.	✓	✓	✓	✓
Utility libraries and development tools	It concerns the availability of a wide range of utility libraries and efficient development tools for a computer language. This can affect the application development time significantly.	G1	C, VB and Java have plenty of utility libraries to use.		✓	✓	✓

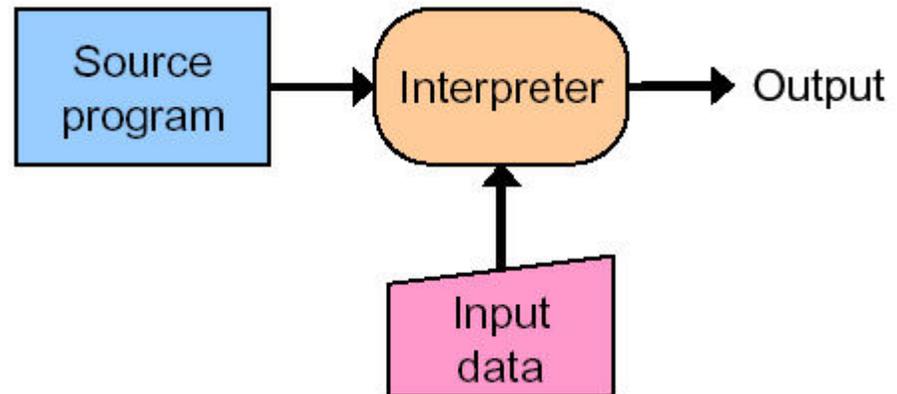
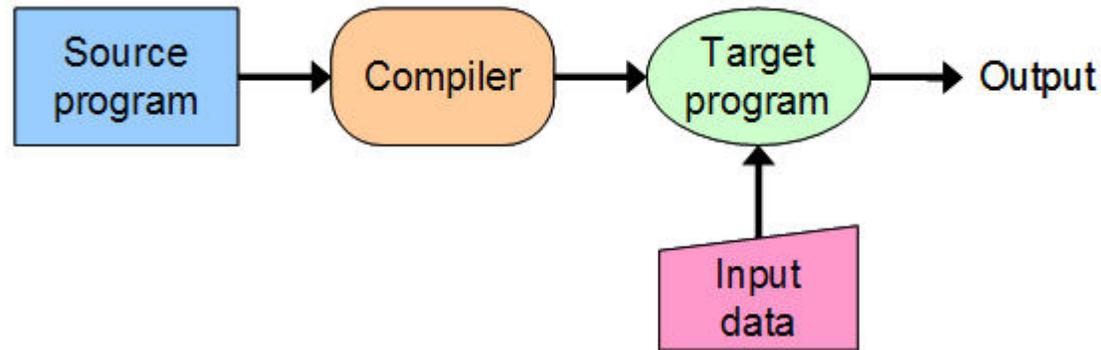
# Topic B2: Language translators and compilers

- Why translator is needed?
- Source code, intermediate code, executable code



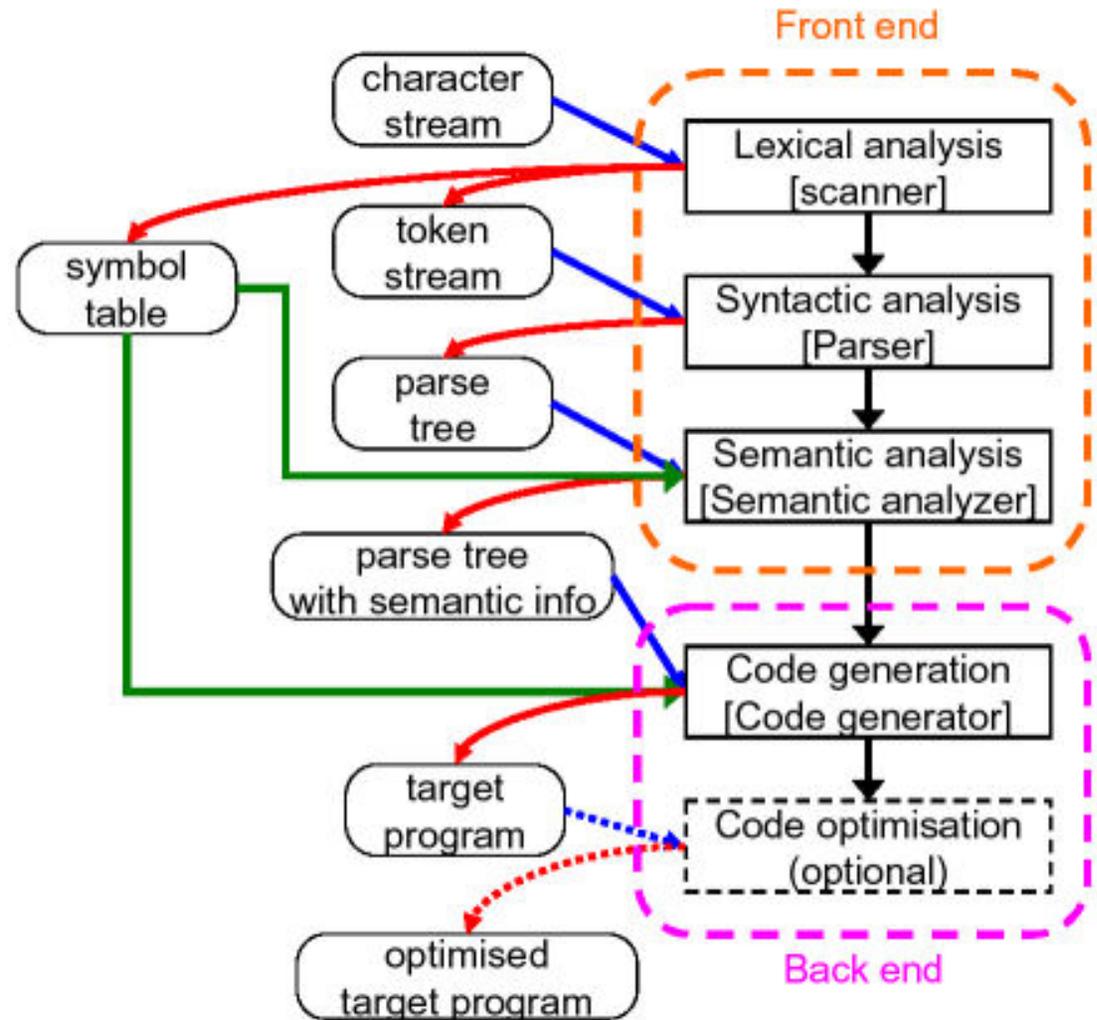
# Compilers and Interpreters

- What a compiler does and what an interpreter does?
- Compare compilers and interpreters



# Overview of phases of compilation

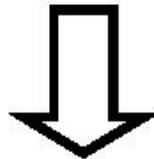
- HKDSE sample paper 2D Q4



# Lexical analysis

- What a lexical analyzer does?
- Character stream from source code → Tokens + Symbols table

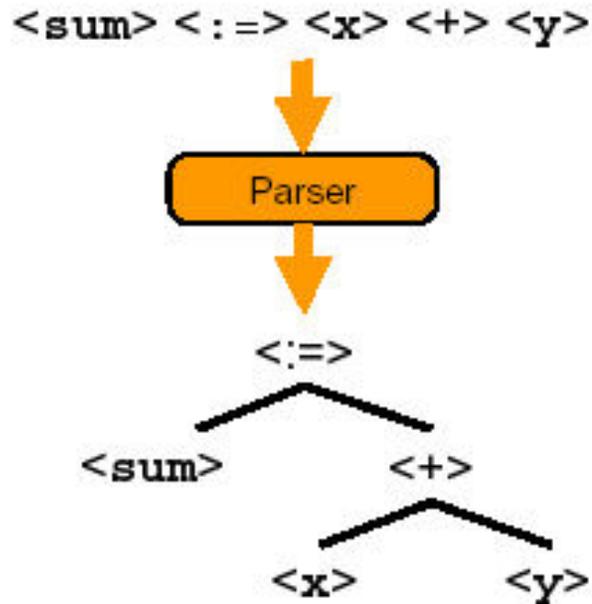
```
program add
[Calculate the sum of 2 integers]
var sum, x, y : integer
start
  input(x, y)
  sum := x + y
  output(sum)
end
```



program	add	var	sum	,	x	,	y
:	integer	start	input	(	x	,	y
sum	:=	x	+	y	output	(	sum
)	end						

# Parser

- What is syntactic analysis?
- Tokens → Parse tree

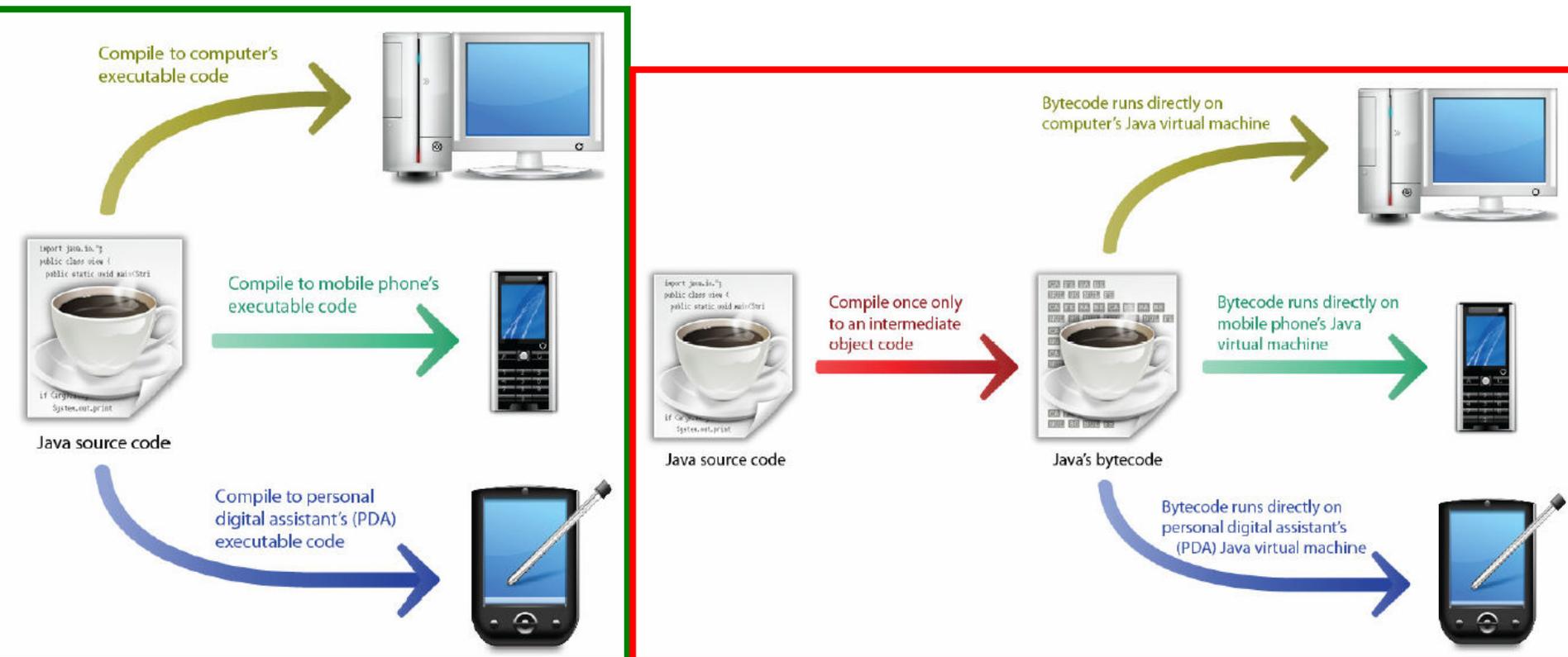


# Semantic analyzer

- What a semantic analyzer does?
- Parse tree + Symbols → Parse tree with semantic information
- Front end

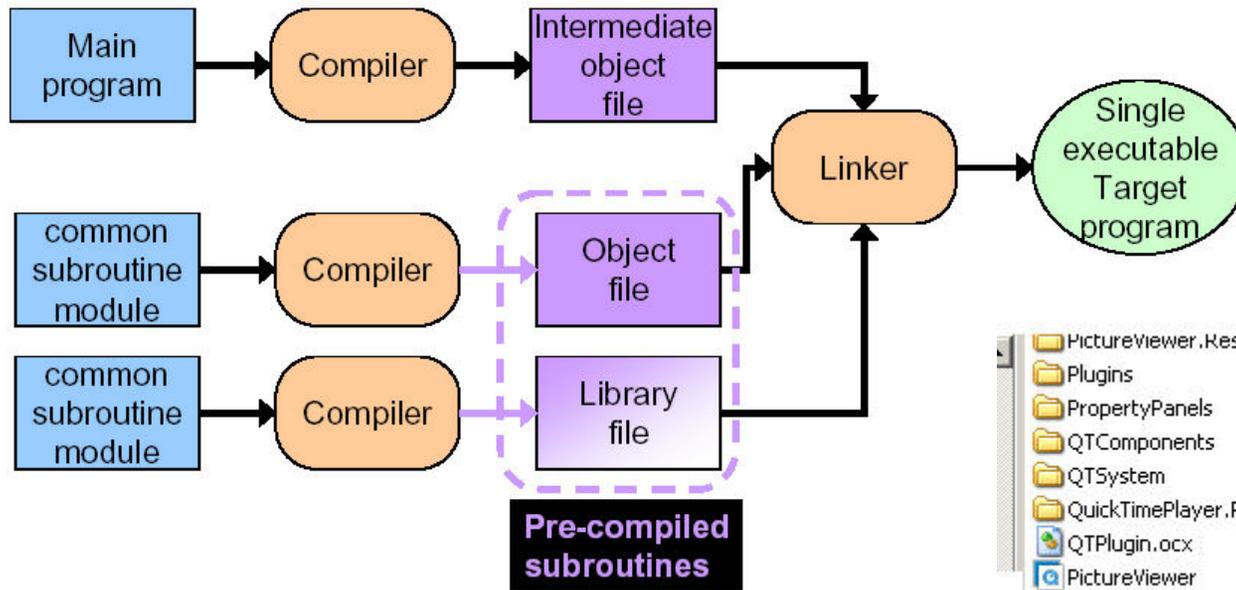
# Back end of a compiler

- Code generation + Code optimization
- Intermediate code (e.g. Java bytecode)
- Java Virtual Machine (JVM)



# Linker and loader

- Functions of a linker and a loader
- Dynamic linking (DLL)



PictureViewer.Resources		File Folder
Plugins		File Folder
PropertyPanels		File Folder
QTComponents		File Folder
QTSYSTEM		File Folder
QuickTimePlayer.Resources		File Folder
QTPlugin.ocx	762 KB	ActiveX Control
PictureViewer	536 KB	Application
QTInfo	764 KB	Application
QTTASK	404 KB	Application
QuickTimePlayer	7,518 KB	Application
QTControl.dll	860 KB	Application Extension
QTOLibrary.dll	788 KB	Application Extension
QTUIPanelControl.dll	344 KB	Application Extension
QuickTime Read Me	10 KB	HTML Document
Sample	19 KB	QuickTime Image
Sample	55 KB	QuickTime Movie

# Assessment Task B #2

## Chapter B2 Language translators and compilers

### Exercise / Quiz

1. (a) The source code of a program written in a high level language must be translated to an object code before the program can be executed on a computer. What is meant by “source code” and “object code”?

---

---

---

---

---

- (b) What are the purposes of a compiler?

---

---

---

- (c) Suggest any TWO reasons why almost all programs are usually compiled before they are sold to customers.

---

---

The End  
Thank you

Pass to “Highlight on Java”

# PROMOTION SESSION ON NSS ICT ELECTIVE –

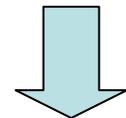
## Highlights on Java

Centre for Information Technology in  
Education, HKU

# Elective Option D: Software Development

- Topic A on “Programming” (47 hours)
  - Choose a programming language from Pascal, C, Visual Basic, **Java**
- Topic B on “Programming Languages” (12 hours)
  - Programming paradigms
  - Language translators and compilers

THE SOURCE FOR JAVA™ TECHNOLOGY  
[java.sun.com](http://java.sun.com)



# Topic A: Java Programming

1	Getting Familiar with the Programming Environment	3 periods	2 hours	
2	Input and Output	3 periods	2 hours	
3	Arithmetic Operations and Functions	4½ periods	3 hours	
4	Selection Structure	6 periods	4 hours	
5	Iteration Structure	7½ periods	5 hours	
6	<b>Programming Modules in Java</b>	9 periods	6 hours	
7	Arrays	6 periods	4 hours	
8	Strings	4½ periods	3 hours	
9	File Handling	6 periods	4 hours	
10	Searching, Sorting and Merging	10½ periods	7 hours	
11	Stacks, Queues and Linked Lists	10½ periods	7 hours	
		Total:	<b>70½ periods</b>	<b>47 hours</b>

Java is an **Object Oriented Language** Yet still allows programming With Modules/Methods !!

# Resource package

- 11 Chapters (A1 – A11)
- 4 Learning tasks
  - A1 – A3: Learning Task A #1
  - A4 – A6: Learning Task A #2
  - A7 – A9: Learning Task A #3
  - A10 – A11: Learning Task A #4
- 4 Assessment tasks
  - A1 – A3: Assessment Task A #1
  - A4 – A6: Assessment Task A #2
  - A7 – A9: Assessment Task A #3
  - A10 – A11: Assessment Task A #4
- Teaching plans for each chapter
- Source programs of examples and practical tasks

# Highlight on Java

- Programming Languages of Choices : Pascal, C, Visual Basic, Java
- Numerous resources for Pascal and C
- Highlights on **Visual Basic** and **Java**

# Highlight on Java IDE - NetBeans

- NetBeans Version 6.7.1 or Later
  - <http://www.netbeans.org>



The screenshot shows a Mozilla Firefox browser window displaying the NetBeans website. The browser's address bar shows the URL <http://netbeans.org/>. The website's header includes the NetBeans logo, a language selection dropdown, and a search bar. A navigation menu contains links for Home, Features, Plugins, Platform, Docs & Support, Community, and Partners. The main content area features a large blue box for NetBeans IDE 6.8, which includes a 3D product box image, a description of the IDE as open-source and free, and a prominent orange 'Download FREE NetBeans IDE 6.8' button. Below this are links to 'Learn More about NetBeans IDE' and 'Get the NetBeans DVD'. To the right, there is a video player for 'Get Tomorrow Today with NetBeans 6.8' and a 'Featured News' section with a link to 'NetBeans IDE 6.9 Milestone 1 Available for Download' and a 'See All News' link.

# Highlight on Java IDE - NetBeans

- Notes on Downloads

Maximal approach --  
Only if disk space  
is not a concern !!

Medium approach --  
If students want to  
explore more on  
Web applications

Minimal approach --  
Readily sufficient  
for all topics/sample  
Programs covered  
in this resource  
Package !!

The screenshot shows the NetBeans IDE Download page for version 6.8. The page includes a search bar, navigation tabs for Features, Community, and Partners, and a section for IDE 6.8 Download. Below this, there are options for IDE Language (English) and Platform (Windows 2000/XP/Vista). A table titled "NetBeans IDE Download Bundles" lists various bundles and their components. The table has columns for Java SE, JavaFX, Java, Ruby, C/C++, PHP, and All. The "All" column is circled in pink, and the "Java" column is also circled in pink. A teal box on the left points to the "All" column, and a green box on the right points to the "Java" column. A pink box at the top right points to the "All" column.

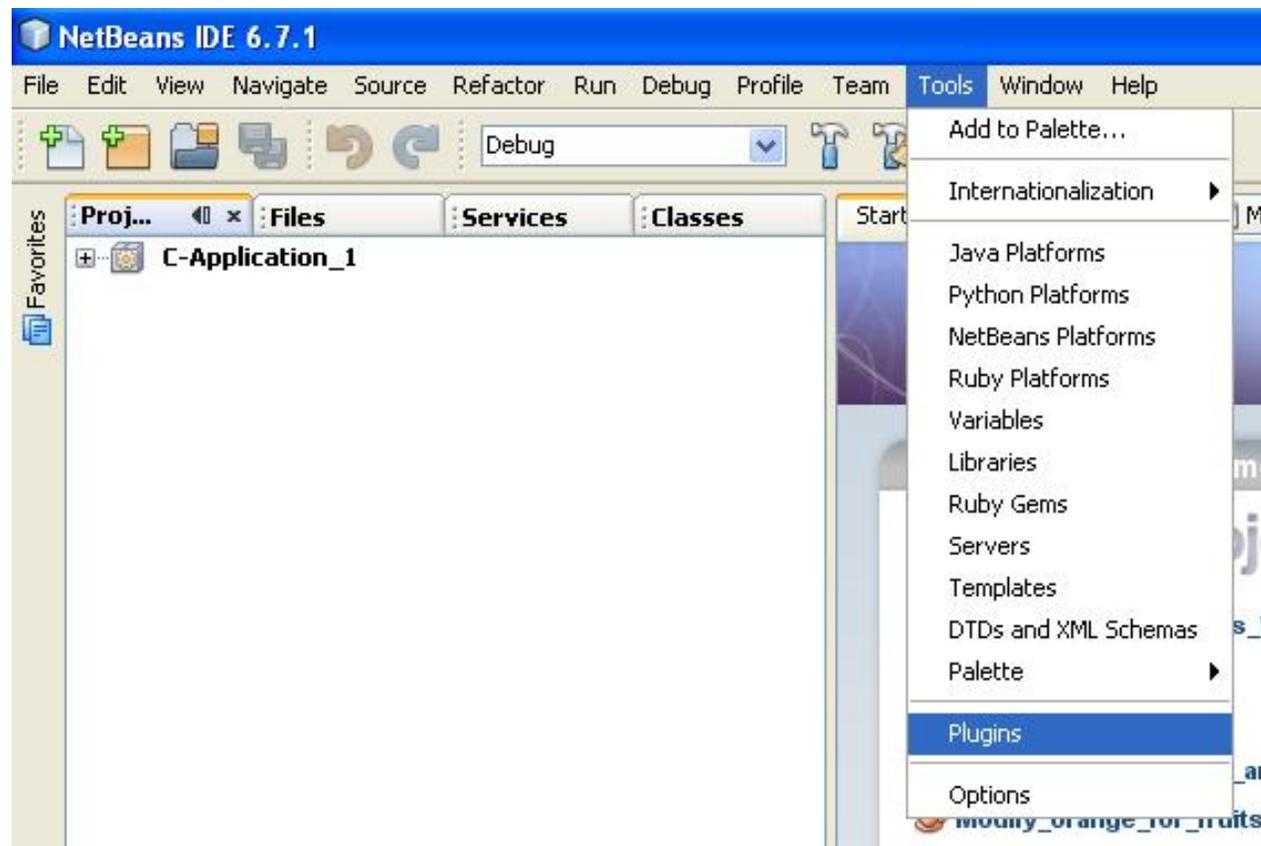
	Java SE	JavaFX	Java	Ruby	C/C++	PHP	All
Java SE	•		•				•
JavaFX		•	•				•
Java Web and EE			•				•
Java ME			•				•
Ruby				•			•
C/C++					•		•
Groovy			•				•
PHP						•	•
Bundled servers							
Sun GlassFish Enterprise Server v3			•	•			•
Apache Tomcat 6.0.20			•				•

# Highlight on Java IDE - NetBeans

- If your download starts with the 'minimal' approach, one can always update with the additional features (such as C++ or PHP) as 'Plug-ins' later.
- In fact, all available features in Netbeans are loaded incrementally as “modules” during program

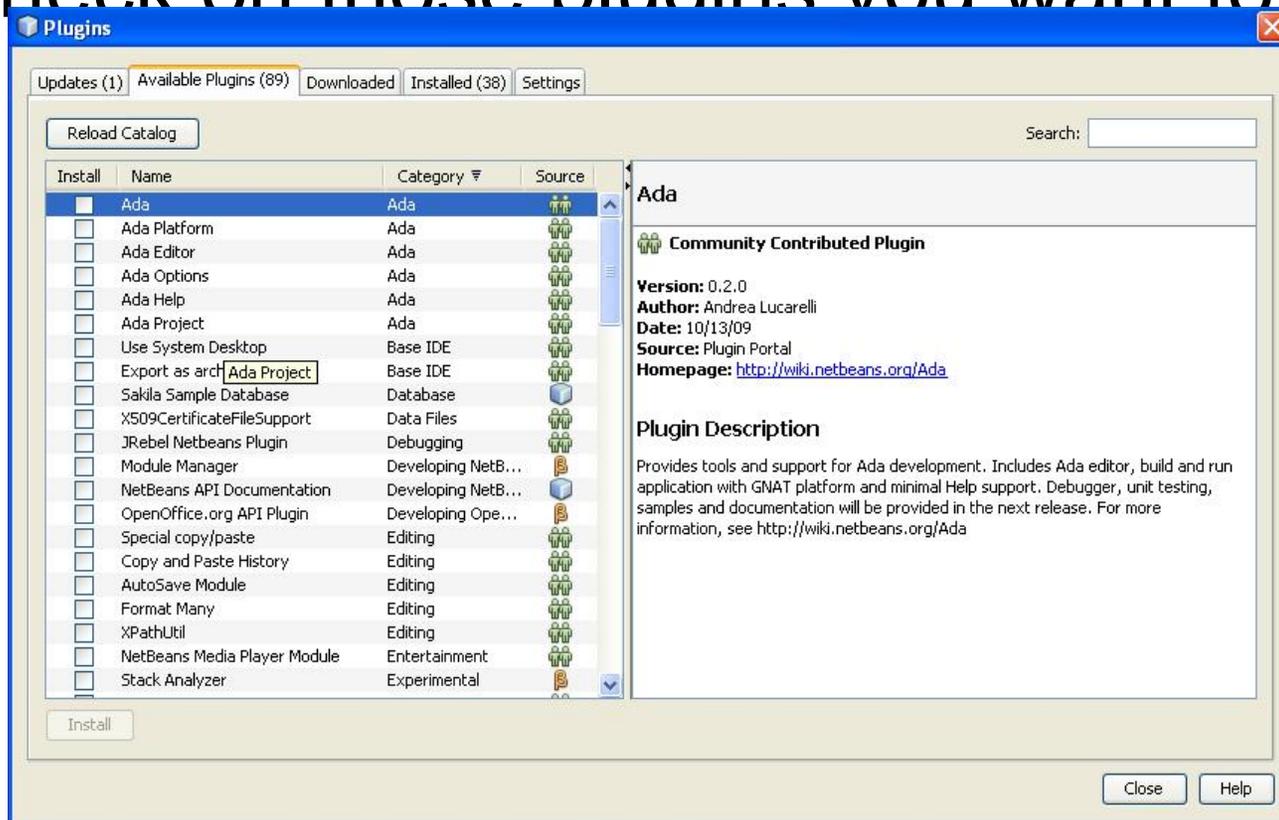
# Highlight on Java IDE - NetBeans

- Updates on Plug-ins:: go to the “Tools” menu -> click on “Plugins”



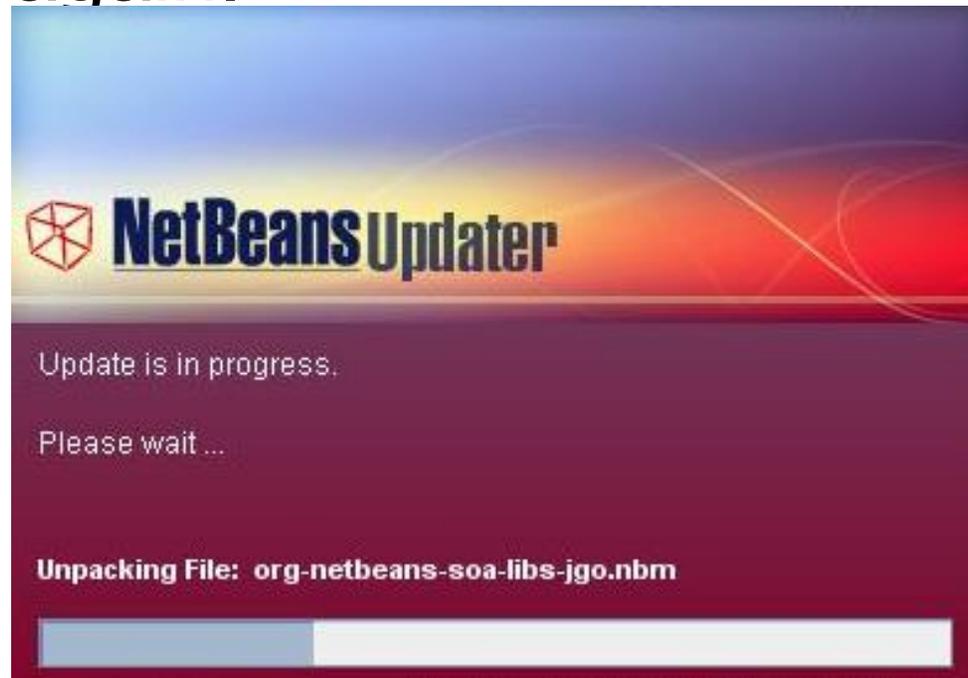
# Highlight on Java IDE - NetBeans

- Next, the IDE “Plugins” window will pop up. Simply click on the “Available Plugins” and check on those plugins you want to install.



# Highlight on Java IDE - NetBeans

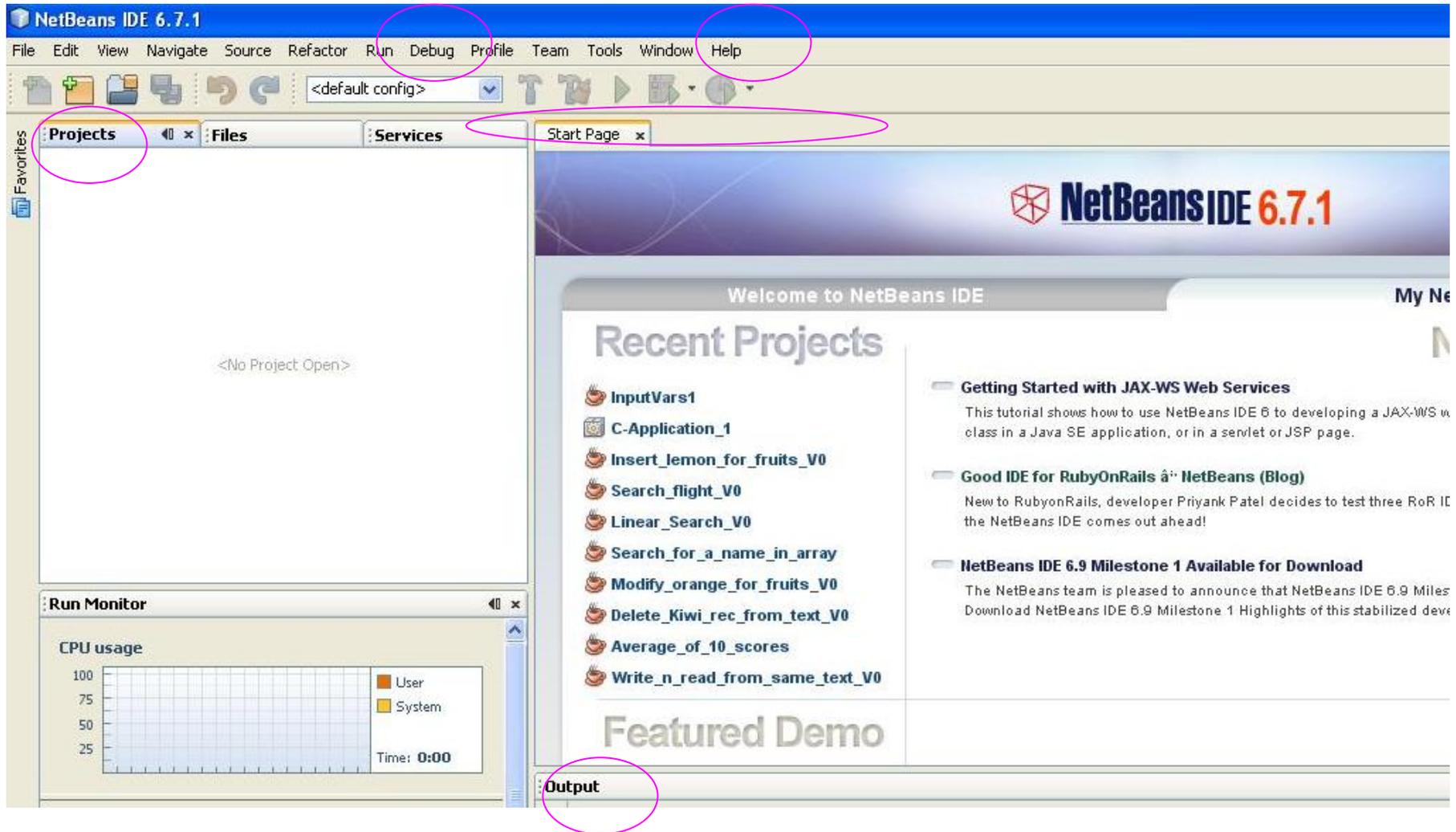
- After the required plugins are downloaded and installed, it will prompt for restarting of the IDE again.



# Minimalist approach

- Students start as soon as possible on some meaningful tasks
  - Understand basic instructions with a flat learning curve
  - Extensive Uses of NetBeans IDE facilities including the **Project Window, Code Window, Output Window, “Debug” function and auto-filling function for suggestions of concerned Java packages, methods and formats of input parameters, and so on.**

# The starting IDE interface..

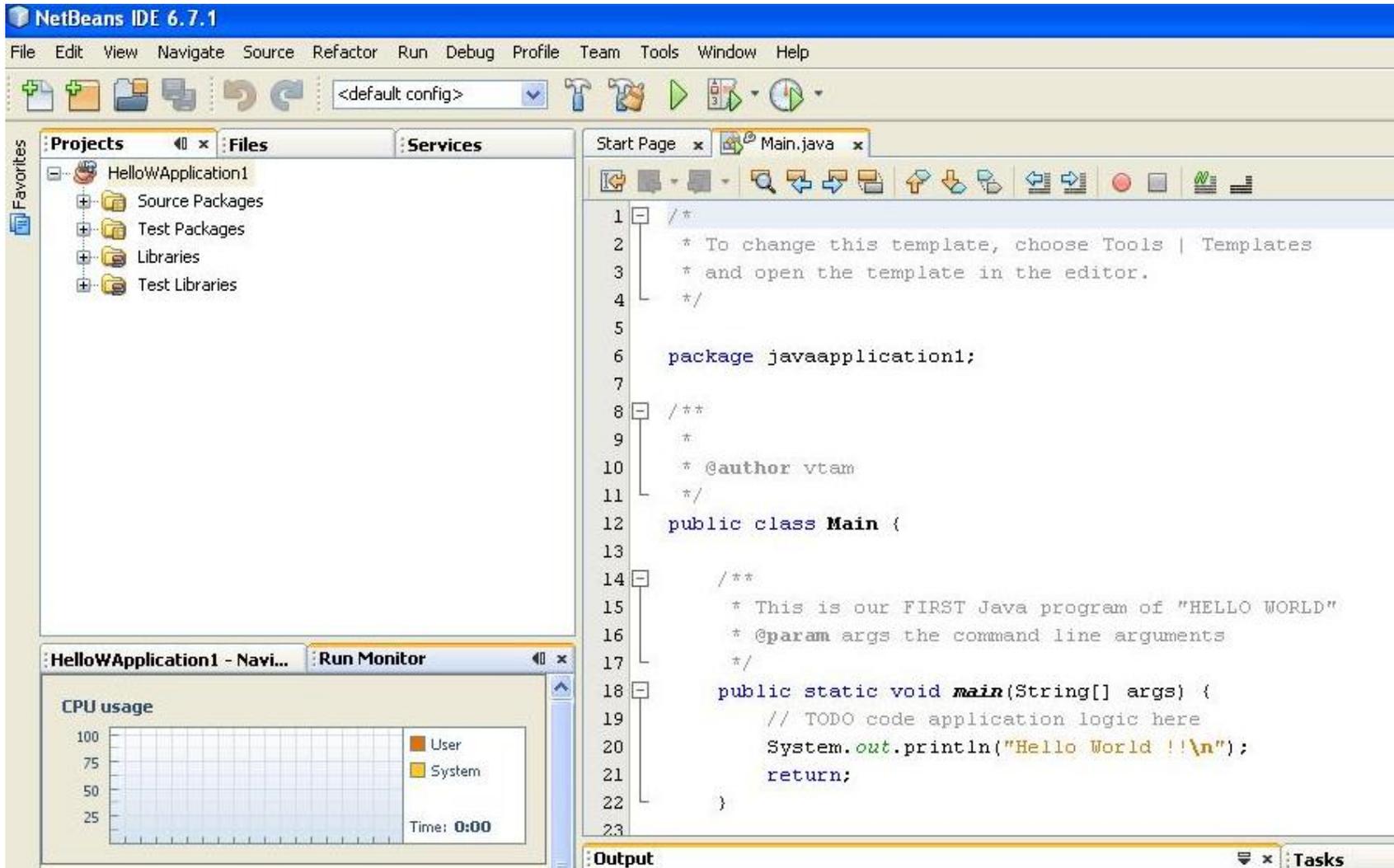


# More on the IDE interface...

The screenshot displays the NetBeans IDE 6.7.1 interface. The **Project Window** (top-left) shows a project named 'JavaApplication1' with a file tree containing 'Source Packages', 'Test Packages', and 'Libraries'. The **Code Window** (center) shows the source code for 'Main.java', including package declarations, class declarations, and a main method. The **Navigator Window** (bottom-left) shows the 'Main' class with a member view containing 'main(String[] args)'. The **Tasks Window** (bottom-right) shows a table of tasks.

Description	File	Line	Location
TODO code application logic here	Main.java	18	...aApplication1\src\javaapplic

# Starting with Simple Java programs for ease of learning and “confidence”..



The screenshot displays the NetBeans IDE 6.7.1 interface. The main editor window shows the following Java code:

```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package javaapplication1;
7
8  /**
9   *
10  * @author vtam
11  */
12  public class Main {
13
14      /**
15       * This is our FIRST Java program of "HELLO WORLD"
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          System.out.println("Hello World !!\n");
21          return;
22      }
23  }
```

The interface also shows a project tree on the left with 'HelloWApplication1' and its sub-packages. At the bottom, there is a 'Run Monitor' window showing CPU usage for 'User' and 'System' over time, and an 'Output' window at the very bottom.

# Starting with Simple Java programs for ease of learning and “confidence”..

- As in most programming paradigms or reference, “Hello World” is often the first and simplest sample program that is sufficient to DEMONSTRATE::
  - The minimal library files/packages that must be included to write a (Java) program;
  - The basic function/method to display/output the “Hello World” message (a constant Java String);
  - How the program interact/communicate with its underlying O.S. platform, i.e. through the main() method of the user-defined class !!

# Starting with Simple Java programs for ease of learning..

The screenshot shows the NetBeans IDE interface. The main editor window displays the following Java code:

```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5
6  package inputvars1;
7  import java.io.*;
8
9  public class Main {
10
11     public static void main(String[] args) throws IOException {
12         // TODO code application logic here
13         int Sum = 0, Num1 = 0, Num2 = 0;
14         String line;
15
16         BufferedReader stdin = new
17             BufferedReader(new InputStreamReader(System.in));
18         line = stdin.readLine();
19         Num1 = Integer.parseInt(line);
20
21         line = stdin.readLine();
22         Num2 = Integer.parseInt(line);
23     }
24 }
```

A callout box with a pink border and blue text says: **Demonstrate the available input methods in Java !!**. An arrow points from this box to the code lines 16-22, which are circled in black. These lines show the use of `BufferedReader` and `InputStreamReader` to read input from `System.in`, and `Integer.parseInt` to parse the input into integers.

The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help), a toolbar, and several panels: Projects, Files, Services, Run Monitor (showing CPU usage), and Output.

# Different Abstraction Levels of Programming..

Application  
Level

Language Level Program

**Compilation**

Machine Level target code  
machine

Program

**Interpreter**

machine

Java

Compilation

Byte-code (.class)

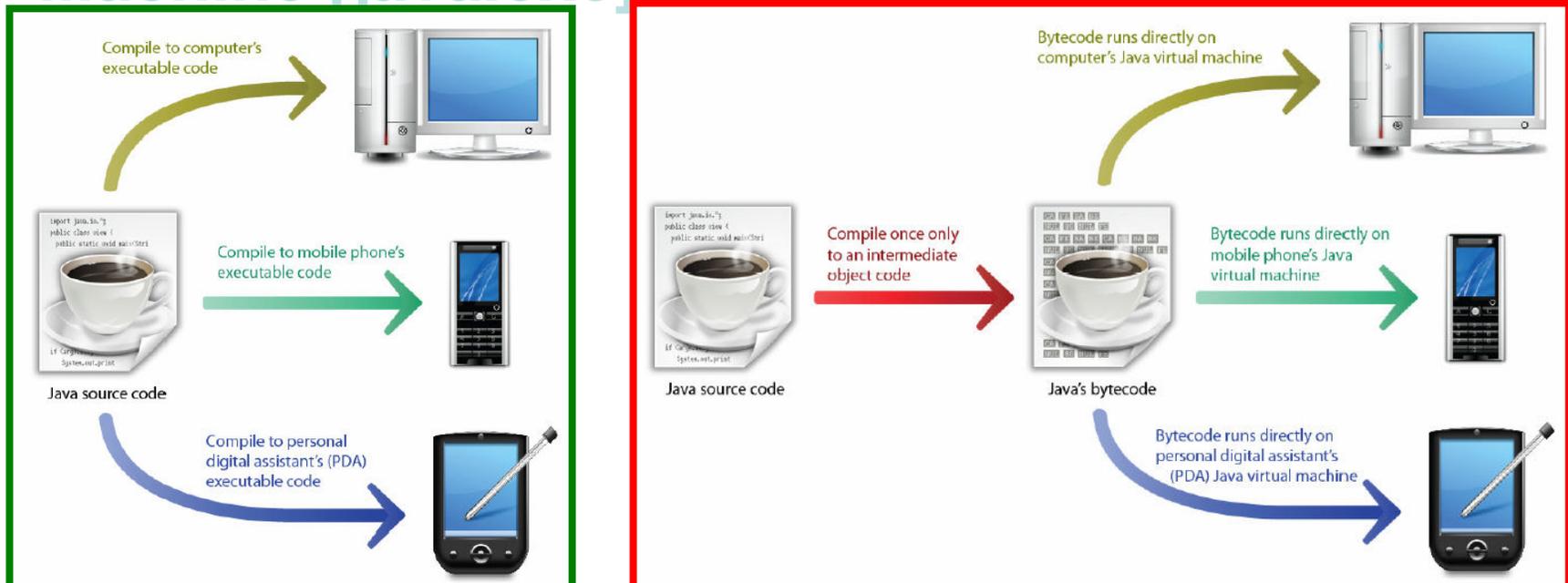
Interpreter (JVM)

Machine

# Java Compiler + Interpreter (JVM)

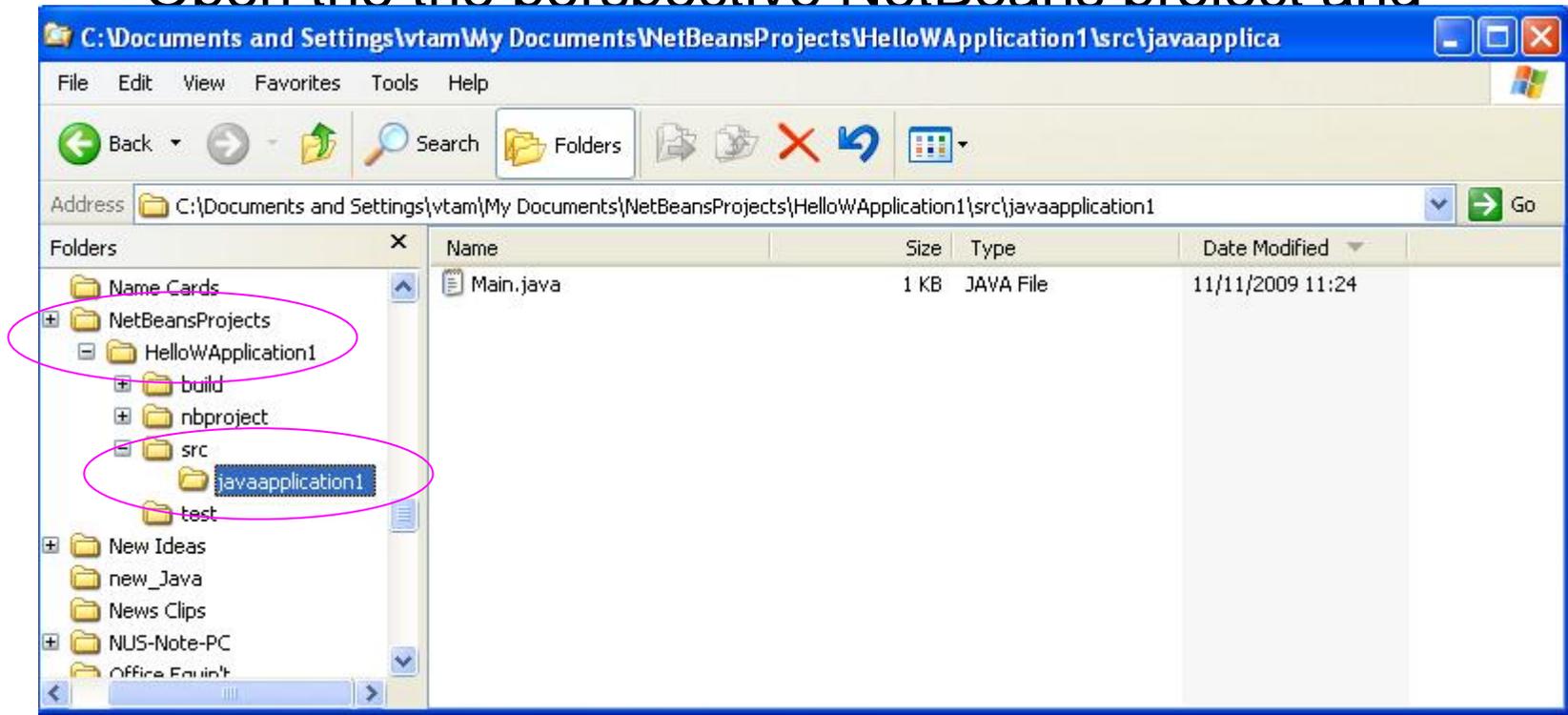
= Portability (or Platform Independence) !!

- Code generation + Code optimization done by the Java compiler [javac.exe in the JDK]
- Intermediate code (e.g. Java bytecode) to be interpreted and executed by the Java Virtual Machine [java.exe]



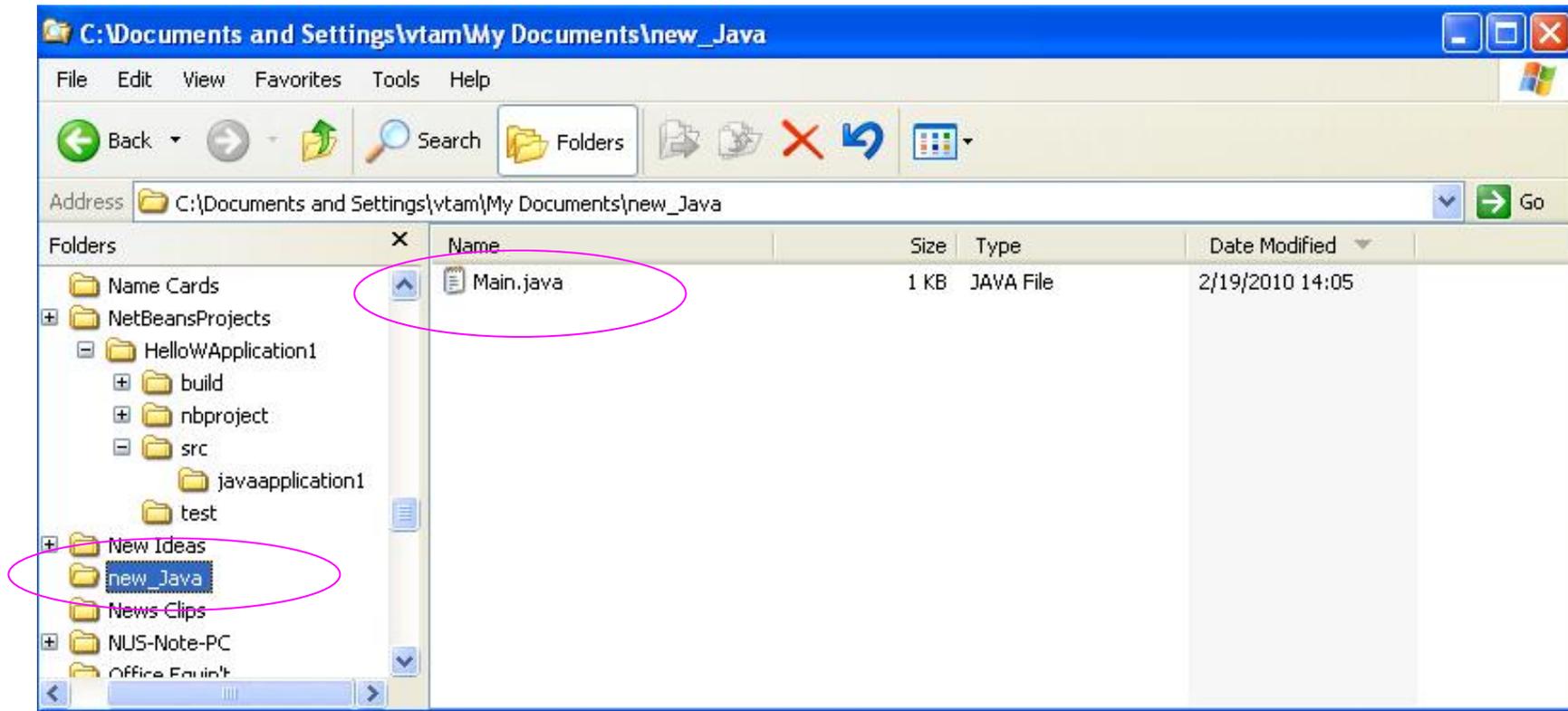
# Demo. of Compilation + Interpretation...(Using Commands)

- Teachers may consider to perform a demo. to the students to clearly illustrate the following steps:
  - Open the the perspective NetBeans project and



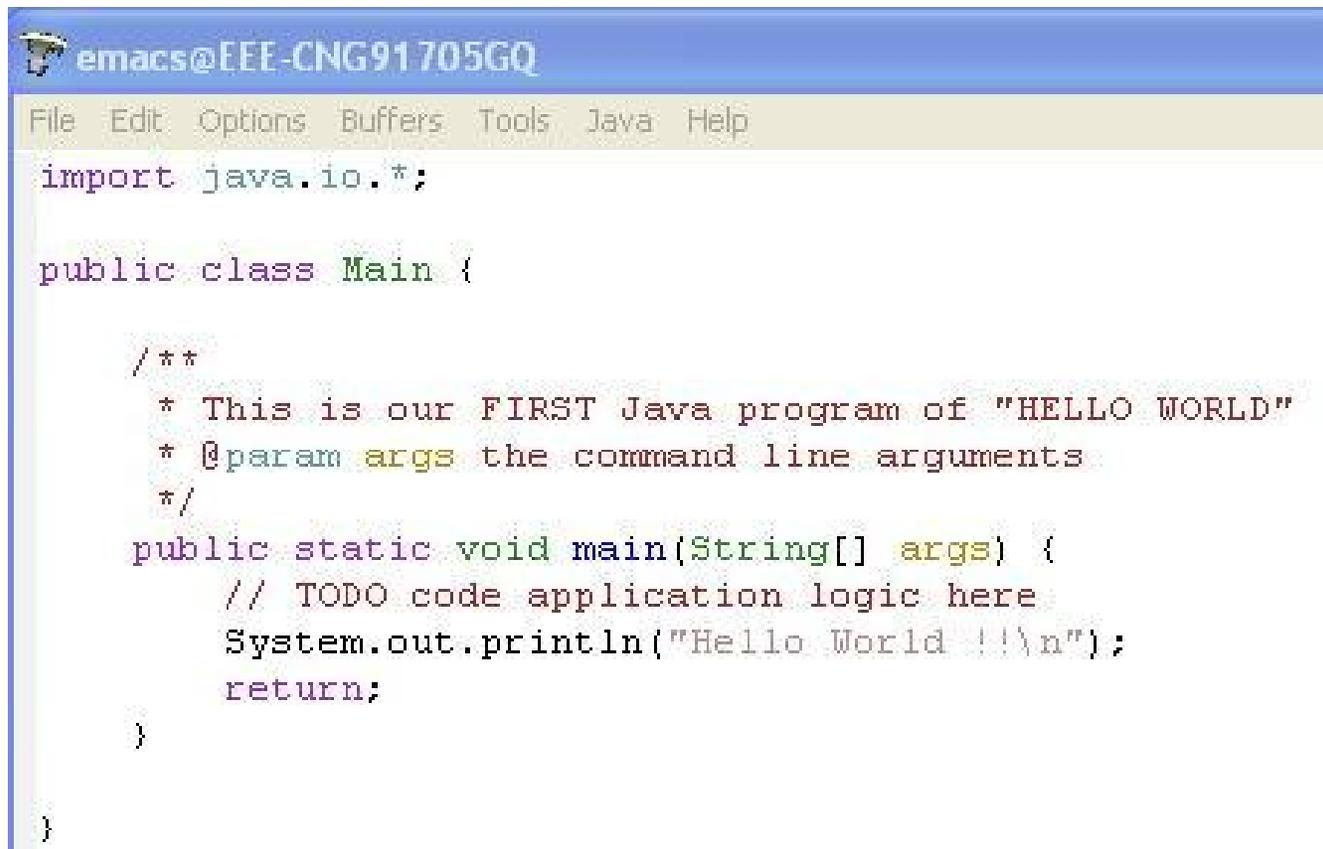
# Demo. of Compilation + Interpretation...(Using Commands)

- Copy the “Main.java” to another new directory;



# Demo. of Compilation + Interpretation...(Using Commands)

- Edit the “Main.java” (using Notepad or your favorite text editor) in the new directory as below, and then SAVE the new version:



```
emacs@EEE-CNG91705GQ
File Edit Options Buffers Tools Java Help
import java.io.*;

public class Main {

    /**
     * This is our FIRST Java program of "HELLO WORLD"
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        System.out.println("Hello World !!\n");
        return;
    }
}
```

# Demo. of Compilation + Interpretation...(Using Commands)

- Now, open a command prompt window and issue commands to compile & execute the saved “Main.java” as follows.

```
C:\ Command Prompt
C:\Documents and Settings\vtam\My Documents\new_Java>javac Main.java
C:\Documents and Settings\vtam\My Documents\new_Java>dir
Volume in drive C has no label.
Volume Serial Number is E450-01D1

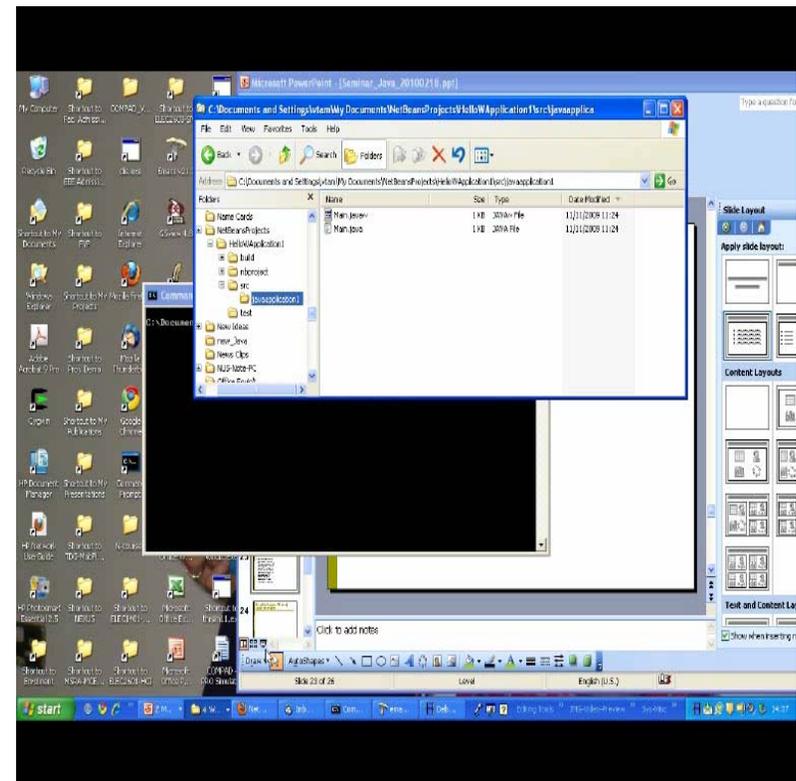
Directory of C:\Documents and Settings\vtam\My Documents
02/19/2010  14:06    <DIR>          .
02/19/2010  14:06    <DIR>          ..
02/19/2010  14:06                417 Main.class
02/19/2010  14:05                319 Main.java
11/11/2009  11:24                455 Main.java~
               3 File(s)              1,191 bytes
               2 Dir(s)  397,600,518,144 bytes free

C:\Documents and Settings\vtam\My Documents\new_Java>java Main
Hello World ??

C:\Documents and Settings\vtam\My Documents\new_Java>
```

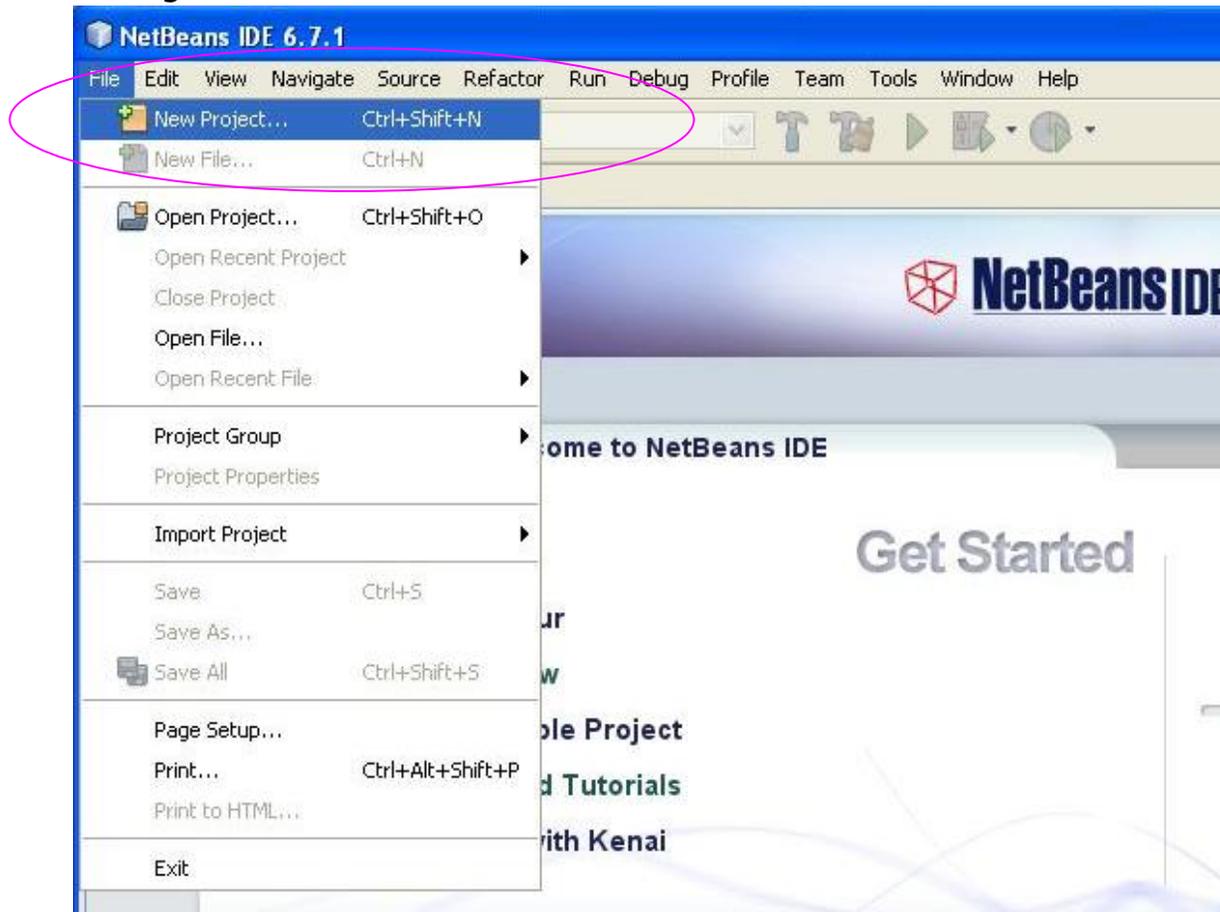
# Demo. of Compilation + Interpretation...(Using Commands)

- Alternatively, teachers may consider to use free video capture tool such as “DEBUT Video Capture” program so as to produce a video file demonstrating the essential steps for issuing relevant commands for Java programs as [HERE](#)



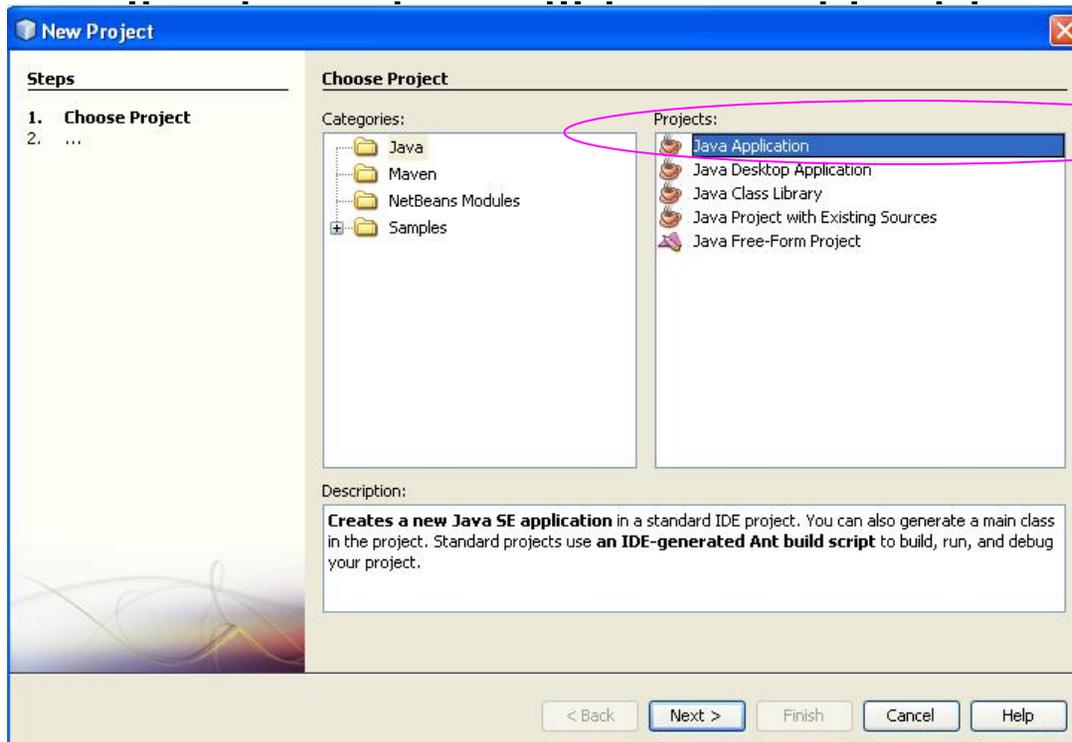
# Console & Swing (or Windows) Applications in Java

- You can begin to write your first program by selecting **New Project** from the **File** menu as below.



# Console & Swing (or Windows) Applications in Java

- The **New Project** dialog box will then be displayed for you to choose the type of applications to be developed. **Java [Console] Application** and **Java Desktop [Swing or Windows] Application** are the two types of source package.

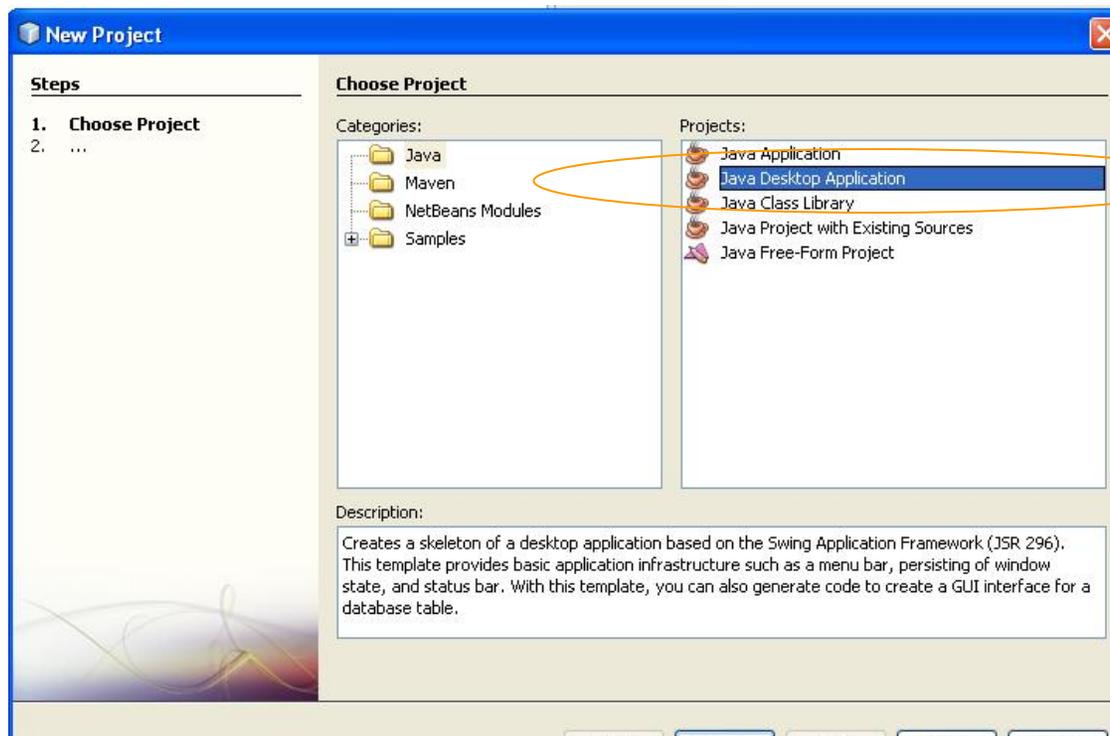


For Writing Java Console Application or Program, i.e. result will be generated in the "Output" Window as the normal console

(ref. Sec. 1.3 of Ch.1)

# Console & Swing (or Windows) Applications in Java

- Alternatively, one can choose the **Java Desktop [Swing or Windows] Application** to produce window-based GUI and related GUI components & methods. However, this approach will be more complicated as it involves deeper understanding of the Swing library (of the Java Foundation Class [JFC]) under the Java2 platform.



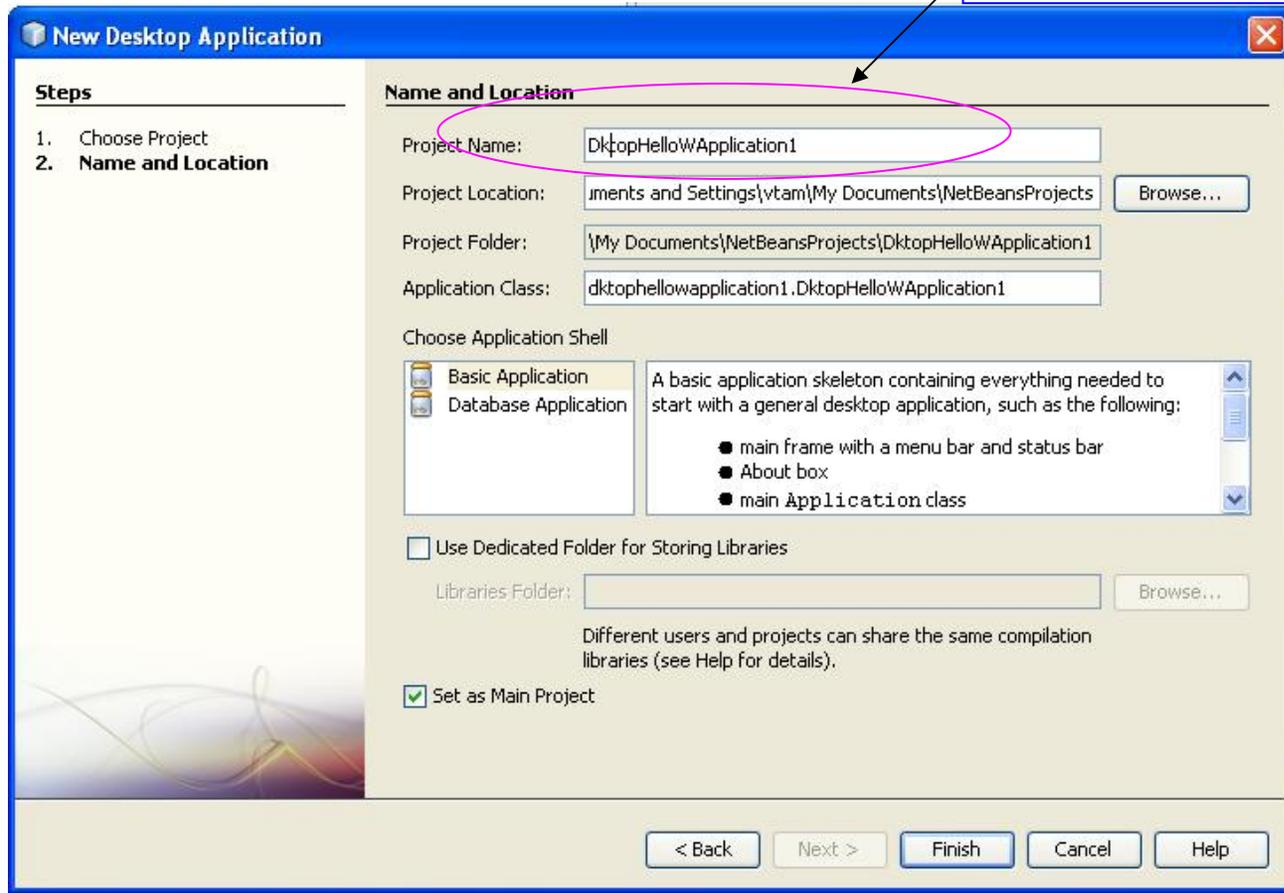
For Writing Java Swing Application or Program, i.e. result will be displayed in a separate window !!

(ref. Sec. 2 of Ch.1)

# Console & Swing (or Windows) Applications in Java

- Here comes some snapshots of the steps involved for generating the **Java Desktop [Swing] Application** of the HelloWorld example covered in Chapter 1.

(1) Fill in the Project Name



# Console & Swing (or Windows) Applications in Java

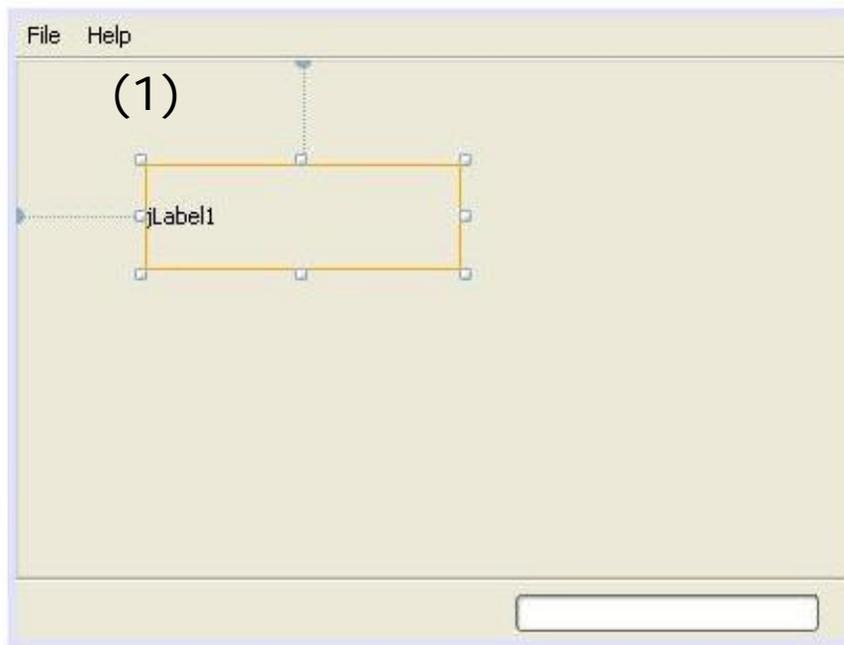
Palette Window

- Use the [Swing] Palette on the right-hand side to add/remove Swing components/controls into the generated **container window** in the center, and also set the properties of the GUI component accordingly.

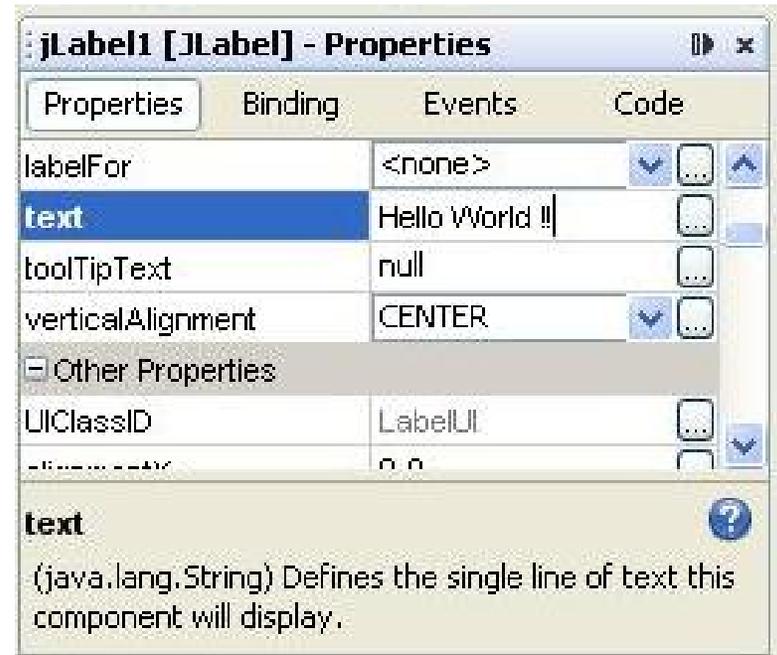
The screenshot displays the NetBeans IDE interface for a Java Swing application named 'DktopHelloWApplication1'. The main workspace is in Design mode, showing a window titled 'File Help' with a text input field. The 'Palette' window on the right lists various Swing components, including 'Swing Containers' (Panel, Split Pane, Tool Bar, Internal Frame) and 'Swing Controls' (Label, Toggle Button, Radio Button, Combo Box, OK Button, Check Box, Button, List). The 'Inspector' window at the bottom right shows the properties of the selected 'FrameView' component, including 'component', 'menuBar', 'statusBar', and 'toolBar'. The 'Navigator' window on the left shows the project structure, including source packages, test packages, and libraries.

# Console & Swing (or Windows) Applications in Java

- Add "Label" and set the text as "Hello World !!".



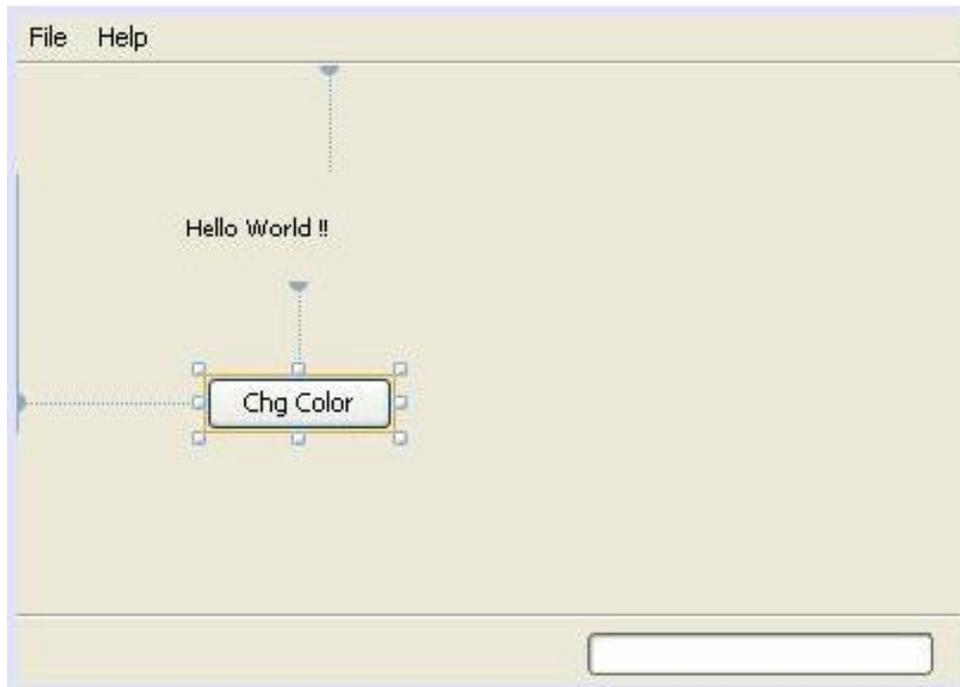
(2)



# Console & Swing (or Windows) Applications in Java

- Add "Button" to change the background color.

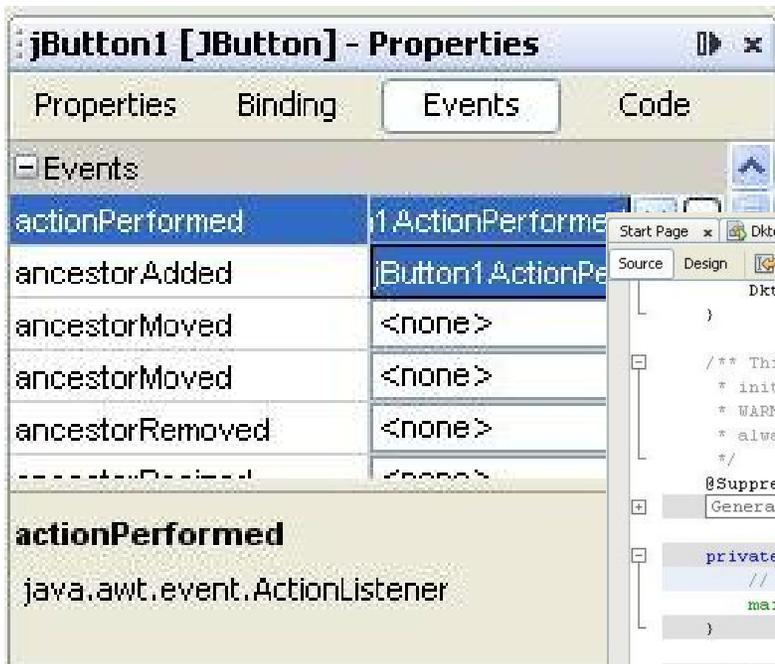
(3)



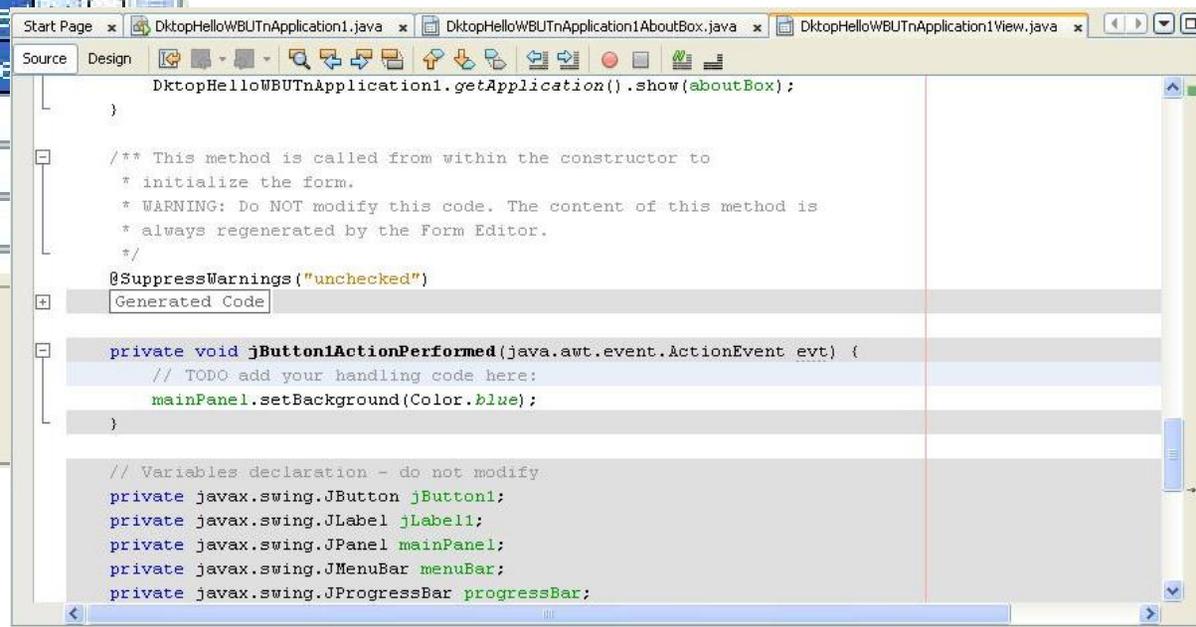
# Console & Swing (or Windows) Applications in Java

- Set properties & write event codes for the “Button” object to change the background color.

(4)

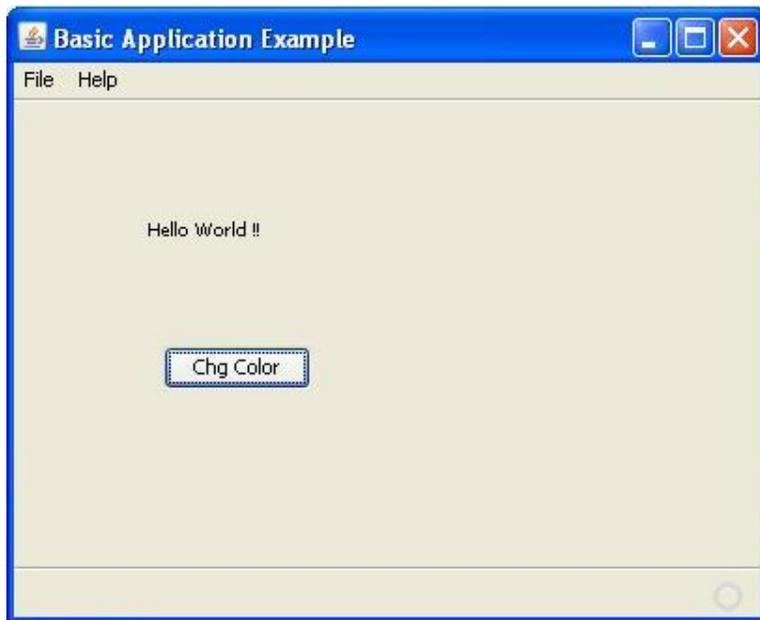


(5)

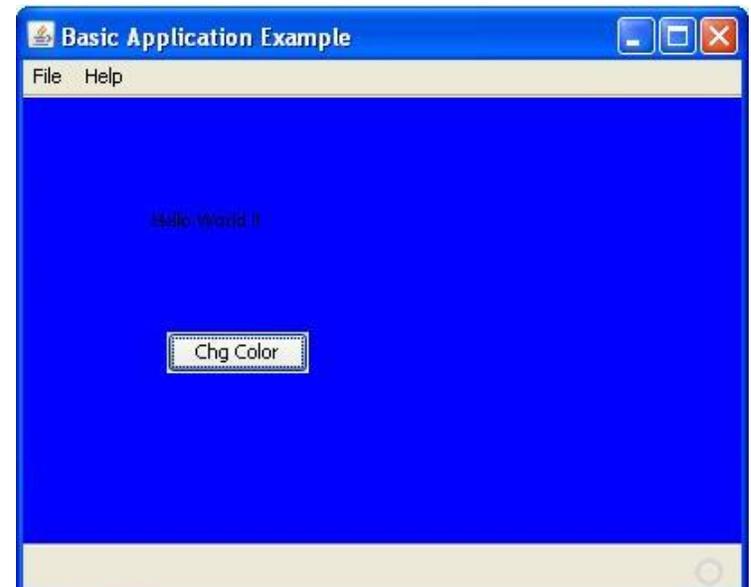


# Console & Swing (or Windows) Applications in Java

- Click on the  on the toolbar, the following window (6) will be displayed. When you clicked on the “Chg Color” button, the background color of the window will be changed to Blue as (7).  
(6)



(7)



# OOP – Classes & Inheritance

- The major distinction between Java (as an OOP language) and imperative programming paradigms such as Visual Basic or C lies in
  - Classes and Objects;
  - Inheritance
  
- >> mainly to facilitate software design & reuses

[ref. Chapter 6 – “Programming Modules in Java”]

# OOP – Classes & Inheritance

- In procedural languages like C, Pascal or Visual Basic, the unit of programming is the function.
- However in Java, the unit of programming is the class from which objects are eventually instantiated (a technical term for 'created').
- Java classes contain attributes (implementing the class data) and methods (implementing the class behaviours). Clearly, the attributes and methods of a class/object are tied together as an inseparable programming unit in Java.

# OOP – Basic Concepts of “Classes”

Real-World Applications::

Template/Model

Actual Item



In Java,

Class

Object

# OOP & Inheritance in Java...

- The basic syntax to define a class in Java is given as follows.

```
class NewClass {  
    . . .  
} // end of class definition.
```

# OOP & Inheritance in Java...

- By the class inheritance mechanism supported in Java, **a newly created class of objects can be derived (or extended) by absorbing characteristics of existing classes and adding unique properties of their own.**

E.g. an object of class '**Convertible**' (to model convertible cars in the real world) will surely have **the basic properties of the more general class 'Automobile'**, **plus a convertible's roof that can go up or down for any convertible car.**

# OOP & Inheritance in Java...

- in Java, one can use the *extends* keyword, and followed by the name of the existing class to inherit from as below:

```
class Convertible extends Automobile {  
    // new fields and methods defining a  
    convertible car would go here  
}
```

End  
- Thank you  
Q & A Session