

Raptor

Thomson Chan
Rosaryhill School
info@chansir.net
36983585


Depository of Curriculum-based Learning and Teaching resources

- <http://www.hkedcity.net/edb/teachingresources>




Depository of Curriculum-based Learning and Teaching resources


- Online learning & teaching tools

 You have not logged in. [Sign In](#) [繁](#) [簡](#) [ENG](#)

[Join Us](#) [Forget Password](#) [Search](#) [Master Calendar](#) [School Map](#) [Site Map](#)

EDB Depository of Curriculum-based Learning and Teaching Resources (Junior Secondary)


 **科技教育**
Technology Education

 [EdBlog \(Comp. Edu.\)](#)

[Home](#) [Learning and Teaching Resources](#) [Professional Development](#) [Related Sites](#) [Online Learning & Teaching Tools](#) [SiteMap](#)

[Home](#)

Strategies & Management




About this site

Features of this Depository of Curriculum-based Learning and Teaching Resources (Junior Secondary Level):

1. The materials introduced in this depository are developed by Education Bureau, HKSARG, schools or sourced from free Internet resources for teachers' free use to facilitate learning and teaching in Hong Kong.
2. The resources are categorised according to the knowledge contexts in the curriculum document.
3. Learning objectives and user tips are suggested in the resources for teachers' reference in lesson planning and preparing learning and teaching materials.

Contact us

For enquiries about this depository, please contact at 3698 3583 / 3698 3608 or seroitetehe@edb.gov.hk / cdoite3@edb.gov.hk

 Curriculum Document

Learning & Teaching Resources of Knowledge Contexts

- [Common Topics](#)
- [Information & Communication Technology](#)
- [Materials & Structures](#)
- [Operations & Manufacturing](#)
- [Strategies & Management](#)
- [Systems & Control](#)
- [Technology & Living](#)

[Resources Search](#)

Depository of Curriculum-based Learning and Teaching resources

- ICT – Programming - [Raptor & C](#)
- Raptor Flowchart Programming Note and Exercise (S3)



Raptor Flowchart Programming Note and Exercise (Chinese version only)

Media: | [Rating & Comments](#)

[Save Resource](#)

Link(s):

- English version
- 01 Variable
- 02 Statement
- 03 Selection
- 04 Iteration

Description:

The package includes presentation files, worksheets, Raptor files for learning and teaching flowchart programming by Raptor.

Source / Author:

Mr. Man K.K. of Lai King Catholic Secondary School

Learning Objective

- Students should be able to acquire knowledge to design and execute algorithms.

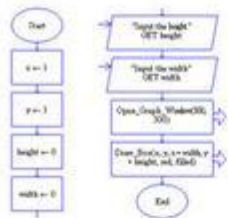
Teaching Suggestions / User Tips

- Teacher can use presentation files to explain concepts in "variables", "statements", selection structure" and "iteration structure";
- Students can complete exercise questions to assess their learning;
- Students can also complete programming exercises to apply programming knowledge.

Keyword(s): 普通電腦 | Computer Literacy | 資訊和通訊科技 | Information & Communication Technology | 程序編寫 | Programming concept | Raptor

Depository of Curriculum-based Learning and Teaching resources

- ICT – Programming - [Raptor & C](#)
- Raptor Flowchart Programming (with C)



Raptor Flowchart Programming

Media: | [Rating & Comments](#)

Link(s):

- + [English version](#)
- + [Topic 1 : Sequential structure](#)
- + [Topic 2 : Selection structure](#)
- + [Topic 3 : Loop structure: FOR](#)
- + [Topic 4 : Loop structure: WHILE](#)

Description:

The package includes different task samples, worksheets, Raptor files and C files for learning and teaching flowchart programming by Raptor.

Source / Author:

Mr. Lau Ka Lun of Lai King Catholic Secondary School

[Save Resource](#)

Learning Objective

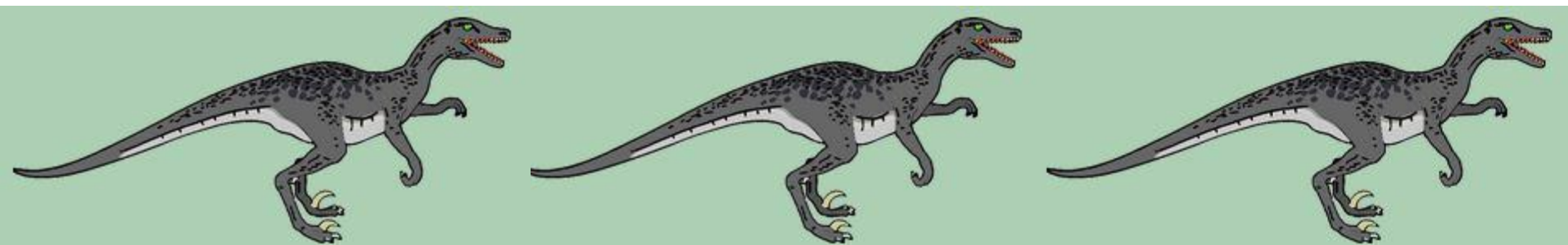
- Students should be able to acquire knowledge to design and execute algorithms.

Teaching Suggestions / User Tips

- Students can read the worksheet first. Each task states the requirement;
- Students can do a "Input-Process-Output" analysis in "a. Problem Analysis";
- Students can refer to own analysis to draw a flowchart draft in "b. Algorithm Design", or use Raptor to program and execute;
- Higher-achieved students can use C or C++ to program and solve the problem in "Programming";
- Students can design different test data and try to execute and debug the programme in "c. Debugging and Testing".

Introduction to Raptor

- A flowchart-based programming environment.
- Help student visualize their algorithms.
- Create visually and executed visually
- Free and not hardware demanding.
- Only need to learn a small amount of syntax.
- Can generate exe file and real code.



Programming Curriculum in RHS

- Taster of programming in S3, up to selection.
- S4 ICT Programming in Core only.
- Raptor + [AppInventor](#) / Scratch
- Online compiler
<http://www.compileonline.com/>

CDI - curriculum

- No flowchart symbols provided in curriculum documents. Drawing Flowchart is needed.
- Pseudo code and Program Flowchart. Translation between flowchart and Pseudo code.
- Simple data types (Raptor: not strong typing) and 1-D array.

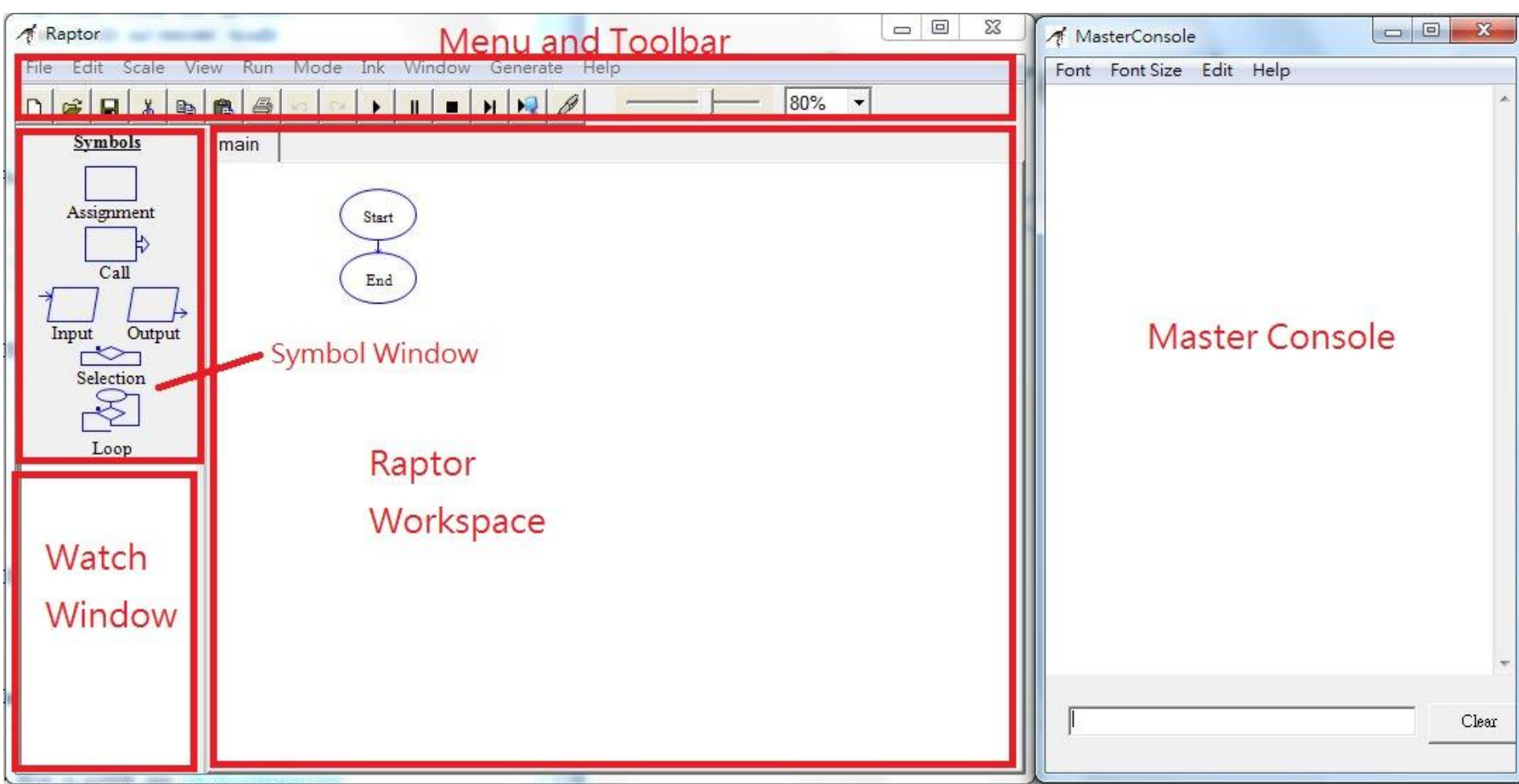
CDI - curriculum

- Basic Control structure. (Raptor: No Multi-way selection and for loop)
- Sub-problems or modules: Use sub-chart in Raptor.
- Dry Run: Trace and test algorithms.

Installation

- <http://raptor.martincarlisle.com>
(raptor_2012.msi)
- Portable version is available
(RaptorPortable_4.0_Revision_6.paf.exe)
- More on portable apps
<http://portableapps.com>

Software environment



Variables

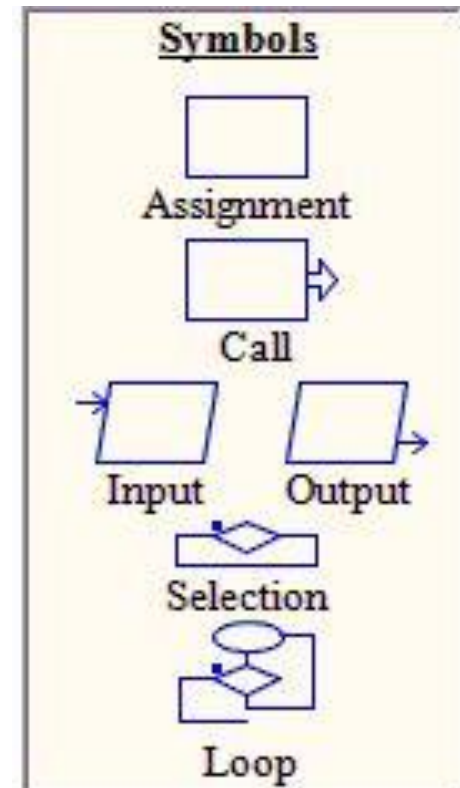
- Name of variable
 - A name must begin with a letter
 - Combinations of letters, digits, or underscores
 - Spaces are not allowed in a name
 - Names are NOT case sensitive
 - Use meaningful name
- Initialize the variable.

Data types

- Not strongly typed.
- Basic Data types: Numbers (real, integer), strings
- Array
 - Index must be positive integer (1,2,3,...)
 - No need to declare the size of array.

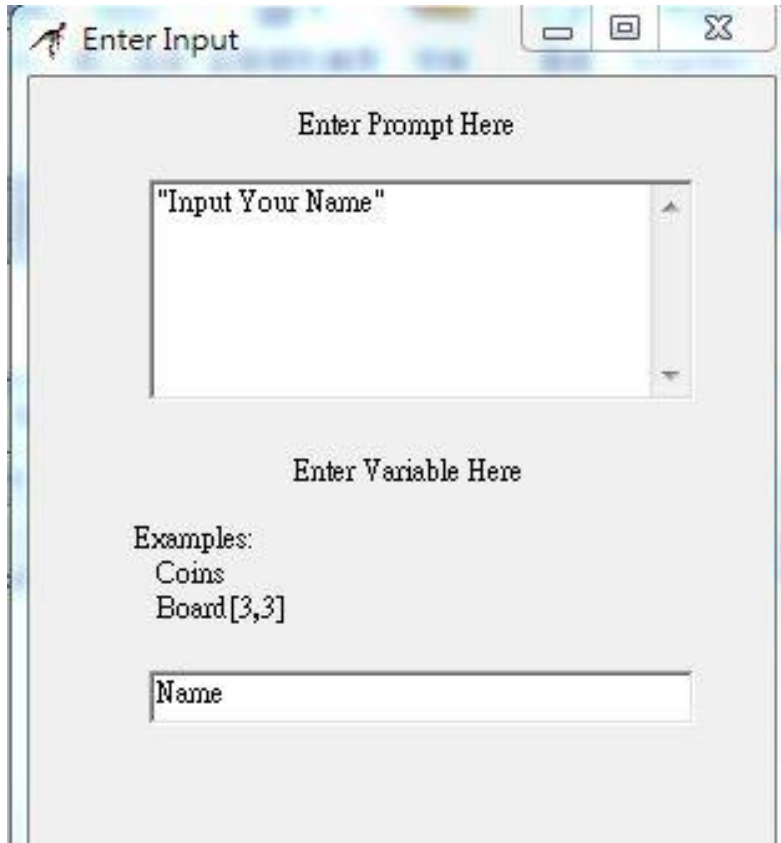
Symbol Window (Statement)

- Six symbols
 - Assignment
 - Input
 - Output
 - Call
 - Selection
 - Loop (Support Pretest and Posttest)

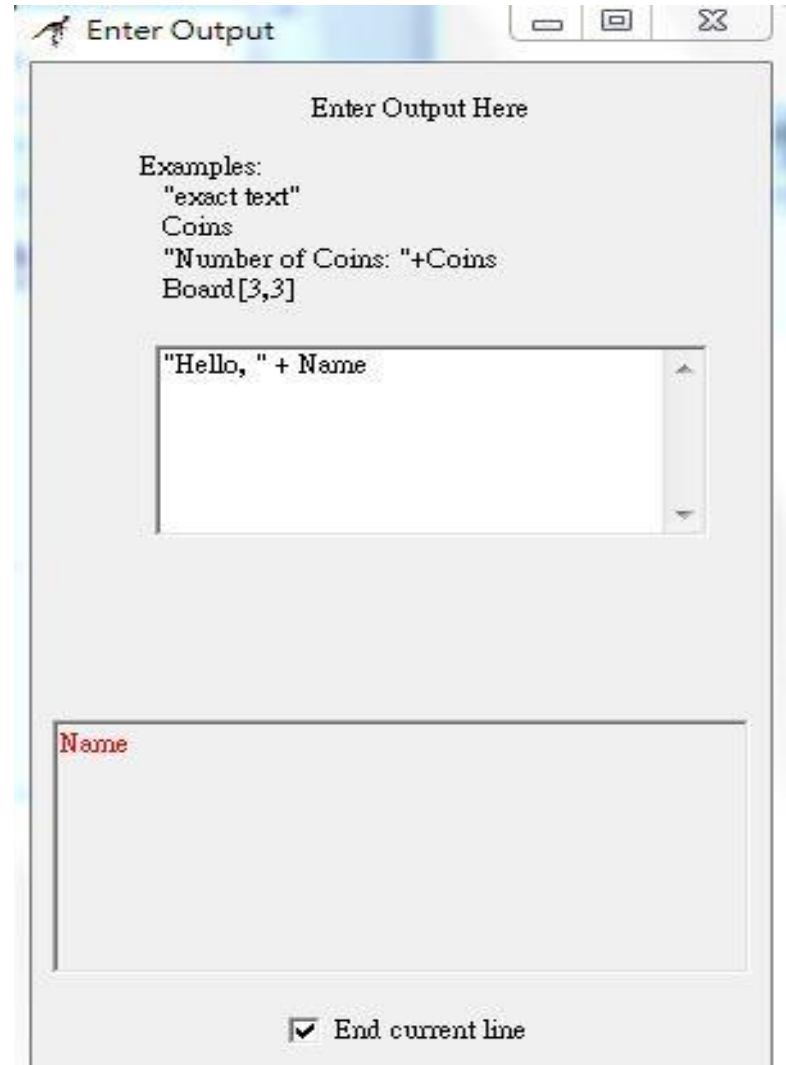


Ex1: Input and output statement

- “ “ string constant
- + Concatenation



The 'Enter Input' dialog box contains a section titled 'Enter Prompt Here' with a text area containing the string `"Input Your Name"`. Below this is a section titled 'Enter Variable Here' with a list of examples: `Coins`, `Board[3,3]`, and a text input field containing the variable `Name`.

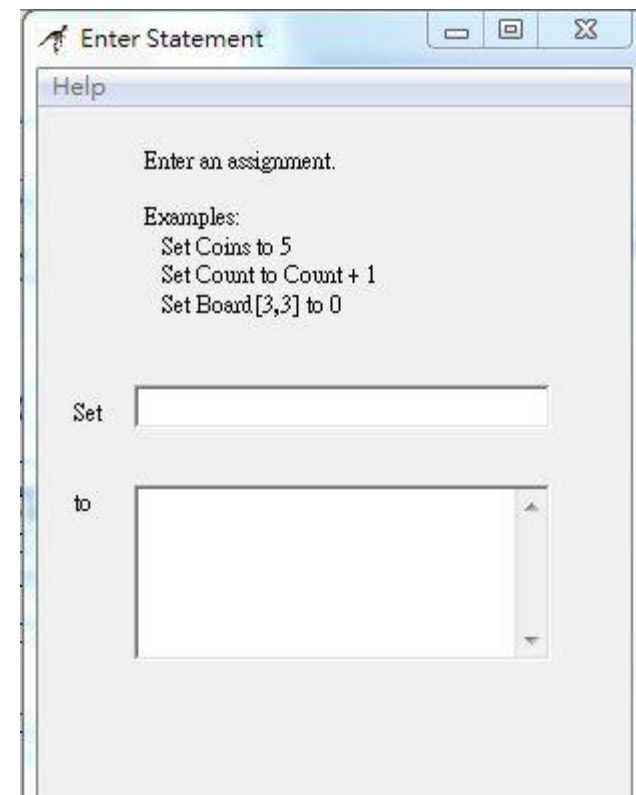


The 'Enter Output' dialog box contains a section titled 'Enter Output Here' with a list of examples: `"exact text"`, `Coins`, `"Number of Coins: "+Coins`, and `Board[3,3]`. Below this is a text area containing the string `"Hello, " + Name`. At the bottom, there is a checkbox labeled 'End current line' which is checked.

Assignment statement

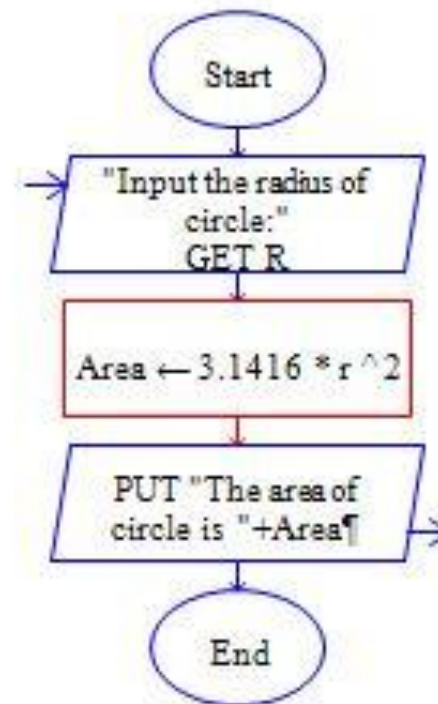
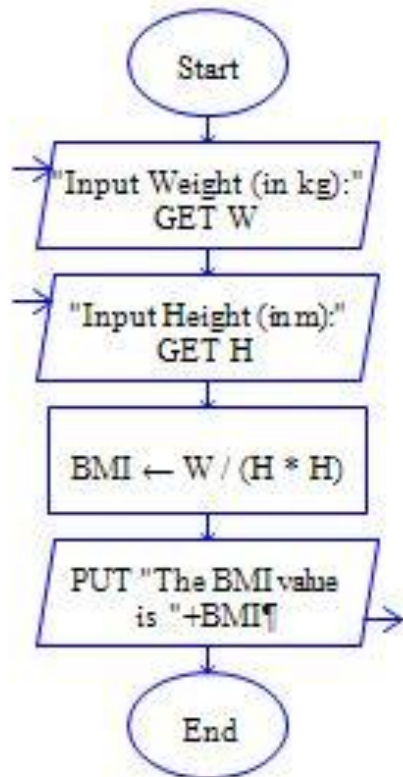
- Set <variable> to <Expression>
- Relational Operator and Logical operator can not be used in <Expression>

Arithmetic operator	+ , - , * , / ^ , ** (exponentiation)
Maths functions	rem, mod (remainder) sqrt, log, abs, sin, cos, tan
Relational operator	= , < , > , <= , >= , != , /= (not equal)
Logical operator	AND, OR, NOT



Ex2:Sequence

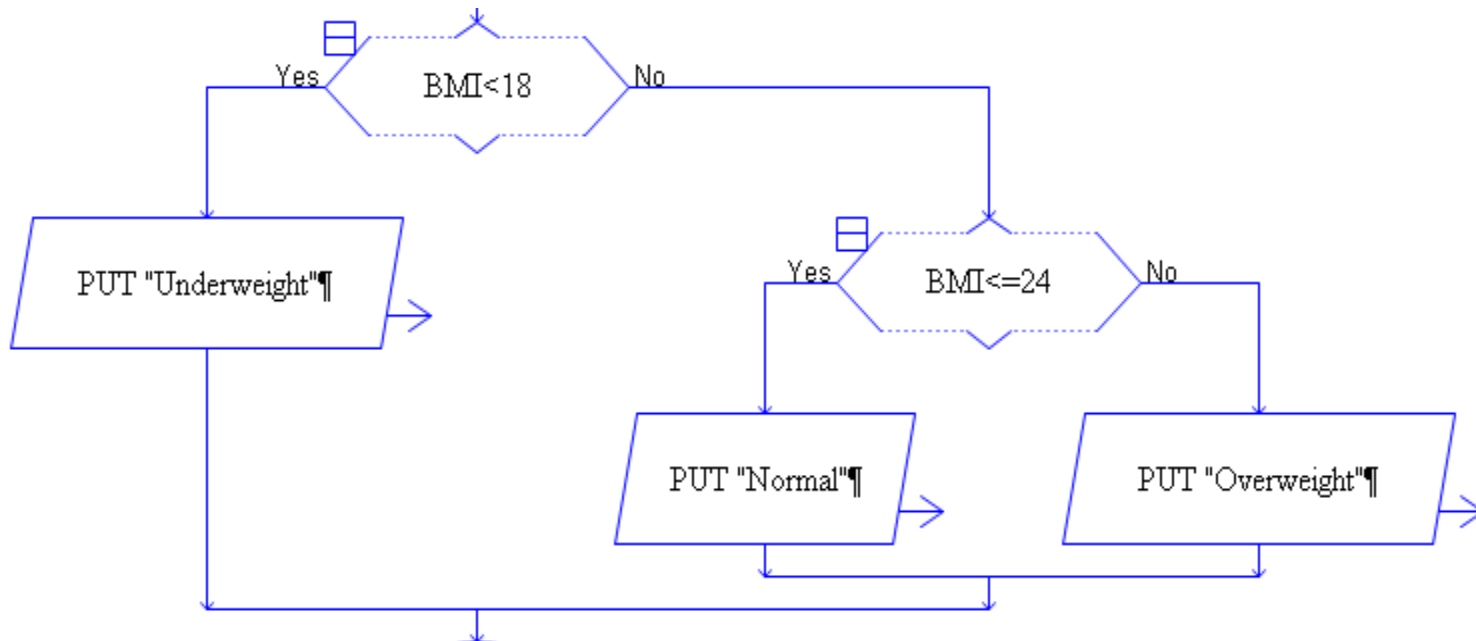
Input	Process	Output
R	$\text{Area} \leftarrow 3.1416 * R^2$	Area
H, W	$\text{BMI} \leftarrow W / (H * H)$	BMI



Selection (if..then..else)

- Ex 2b

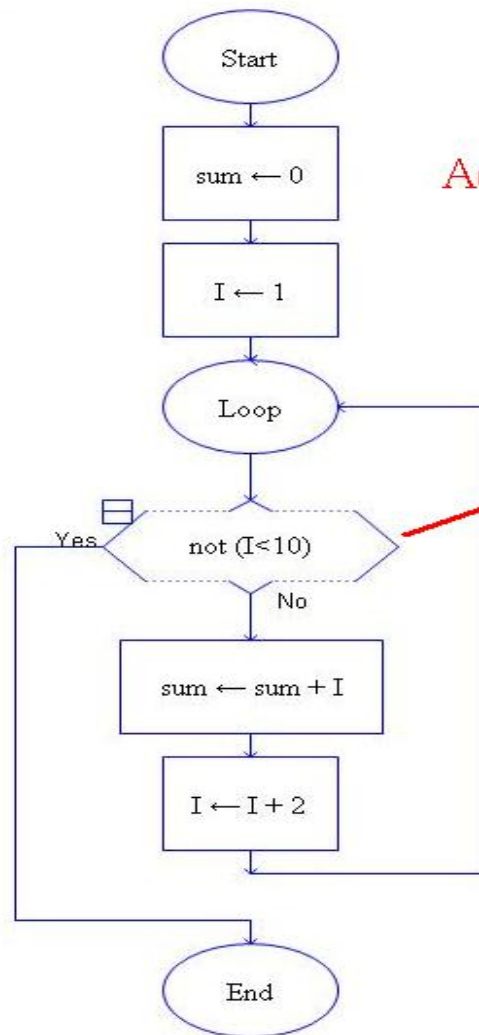
BMI value	Output
Below 18	Underweight
18 - 24	Normal
Above 24	Overweight



Iteration (pre-test / Post-test /For)

Pre-test	Post-test	For loop
execute (0..N)	at least once. (1..N)	No of times known.
While <condition> do <Loop body>	(Raptor) Repeat <loop body> Until <condition> Or Do <loop body> While <condition>	For count=M to N do <loop body>

Ex 3a: Pretest (while .. Do)



Add a NOT before the condition

sum ← 0

I ← 1

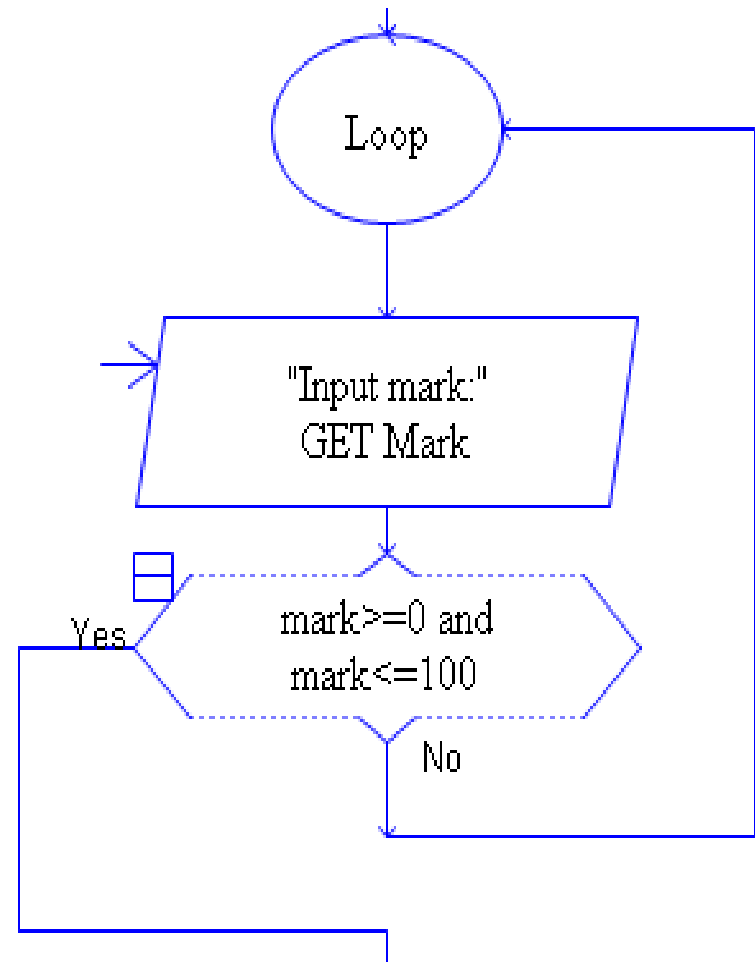
while (I < 10) do

sum ← sum + I

I ← I + 2

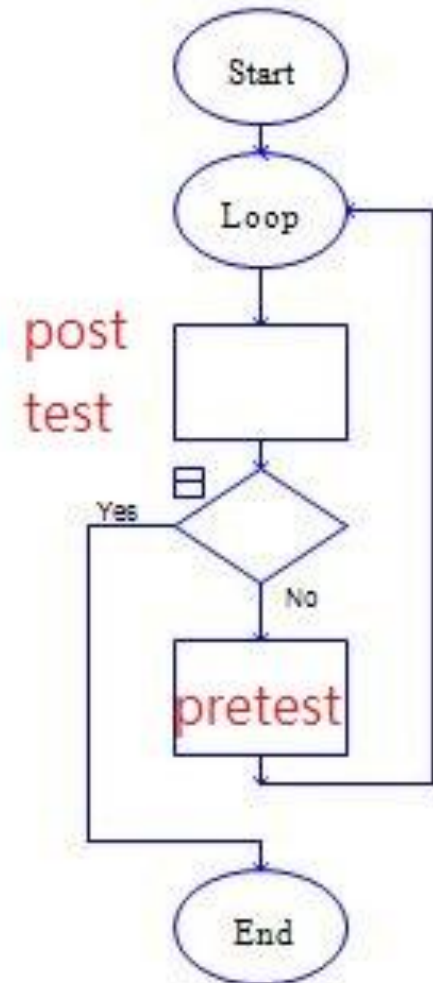
Ex 3b: Post-test (Repeat..until)

- Validation
- $0 \leq \text{mark} \leq 100$
- Amend the condition when writing pseudo code using do..while.



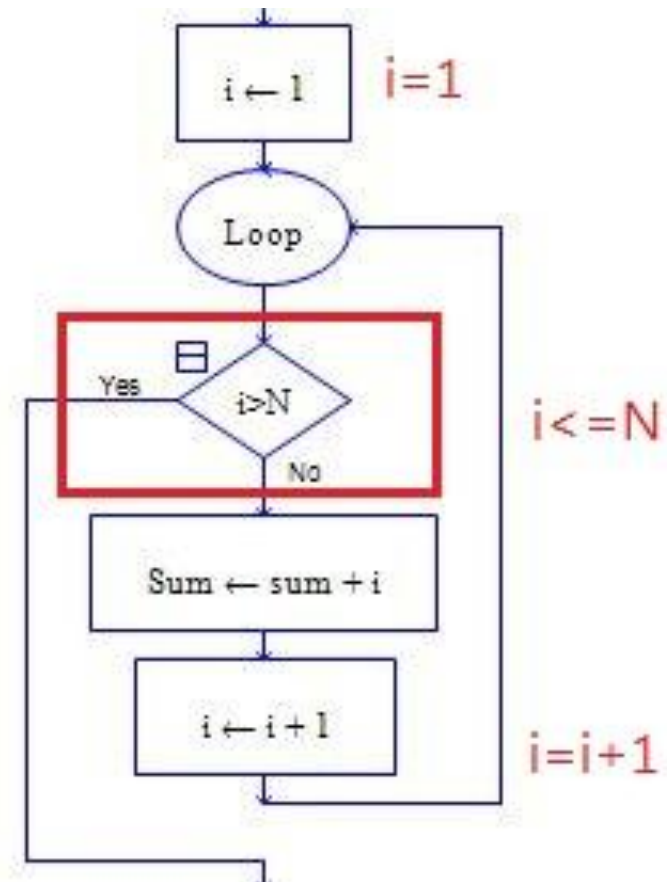
Avoid using unstructured loop

- Either use Post-test statement or pretest statement.



Iteration (Simulate For Loop)


- Calculate $1+2+\dots+N$
- In C++
 `cin>>N;`
 `int sum=1;`
 `for (int i=1; i<=N; i++)`
 `{sum=sum+i;}`
 `cout<<sum;`
- Prefer to use Scratch or other tools.



Sub-chart / Procedure

- Build in procedure support variable passing.
- Novice mode (create sub-chart)
 - Variable are shared (All are Global)
 - Do not support passing of parameters.
- Intermediate mode (create procedure)
 - support parameter passing and recursion.
- [Demo Video](#)

Other features

- Adding Comments
- Run step by step (F10 or icon )
- Toggle breakpoint.
- Generate executable (exe file)
- Menu – Ink (Draw / Erase)
- Drawing by calling build in procedure.

Generate Source code (Ada, C#, Java, C++)

- C++ are case sensitive.
- Change all variable into small letter. (i, sum)
- Add “output sum” at the end.
- Generate C++ for ex3a.rap
- Add data type (int) for i and sum
- Copy and paste to the URL
http://www.compileonline.com/compile_cpp_online.php
- [Demon Video](#)

Viewing student's work

- Students are required to submit *.rap
- HELP : Show Log, Count Symbols
- Command Extract_times_all in Master Console.
- [Demon video](#)