Java

Introduction to Programming Principles

Study as if you were to live forever. Live as if you were to die tomorrow.

- Isodore de Seville

Programming Languages

- A *programming language* specifies the words and symbols that we can use to write a program

- Hundreds of languages have been developed
- Fortran, COBOL, Pascal, C, C++, Ada
- Java one of the more recent additions
  - Java was created by Sun Microsystems, Inc.
  - It was introduced in 1995 and has become very popular
  - It is an object-oriented language

Languages used in organisations

- C++ (18.00%)
- Java (16.70%)
- HTML/Javascript/ASP/XML (13.60%)
- Visual Basic (9.8%)

Source: META Group Inc
2003 Worldwide IT Benchmark Report

Six steps in problem solving

- The purpose of writing a program is to solve a problem

1. Understand the problem
2. Determine how we can test our solution
3. Design a solution
4. Test the design
5. Write the program(s)
6. Test the solution and fix any

1. Understand the problem

- This is systems analysis
- Avoid assumptions and ambiguities

- You will be given problems that you must solve by writing programs in Java
- Ensure you completely understand what the problem is
- Do exactly what you are asked

Avoid ambiguities

2. Determine how to test the solution

You can’t do this until you understand the problem

- Develop a test plan
  - This contains a number of test cases
- A test case contains
  - input
  - expected output
  - a column to compare expected output with actual output after the program is completed
- Develop a sufficient number of test cases to be satisfied that the program works over a range of values

Test Plan

Input variable name or prompt	Input value	Expected output	Actual output
Enter fahrenheit temperature:	300	148	
	212	100	
	100	37	
	99	37	
	0	-17	
	-1	-18	
	-40	-40	

3. Design a solution

Again, before writing a single line of Java (or other) code

- Develop a set of precise steps (ie an algorithm) that describe exactly:
  - the tasks to be performed
  - the order in which they are to be carried out
- For this we use pseudocode
  - Enforces concentration on the problem not the syntax of the language

Algorithms

- Knitting algorithm

SLEEVES

Size 4 mm (8) needles

Working in St. St.

Pick up 70 (76, 82, 88, 102, 114, 118) Sts. around armhole edge

Work 10 (10, 10, 10, 12, 2, 0, 0) rows in St. St.

Next row K.2. K.2. Tog. T.B.L. K. to last 4 Sts. K.2 Tog. K.2.

- Cooking algorithm?
- Musical algorithm?

Programming Algorithm

Requirements Specification

Write a program that displays a prompt for a user to enter a temperature in Fahrenheit. Convert the temperature to Celsius and display it on the screen.

ConvertFahrenheit

Prompt operator for fahrenheitTemp

celciusTemp = (fahrenheitTemp – 32) * 5 / 9

Display celciusTemp

END

Pseudocode

- This algorithm is written in a structured form of English called *pseudocode*.
  - It is not complete English sentences
  - Nor is it the code of any programming language

- Pseudocode is an outline of a program that can easily be converted into real program statements
- It is easy to read and write
- Like a bulleted “to do” list of statements
- Precise syntax is not so important

Pseudocode

- Name the algorithm
  - The name should describe the process
  - Often start with a verb
  - Capitalise words and leave no spaces eg. ConvertFahrenheit
- Indicate the end with the word END
- Each line has a single processing step
  - Data items need meaningful names
  - start in lowercase with capitals for starting letters in subsequent words (no spaces) eg celciusTemp

4. Test the design

A very important step

- It is easier to detect logic errors in the pseudocode than in the actual program code
- “Desk check” the algorithm
  - Walk through the algorithm with the test cases
  - You act as the computer
  - Keep track of variables on a sheet of paper

5. Write the program(s)

Write the program code in the appropriate language (Java)

- Avoid the impulse to start here!!
- Only begin coding after the previous steps have been completed

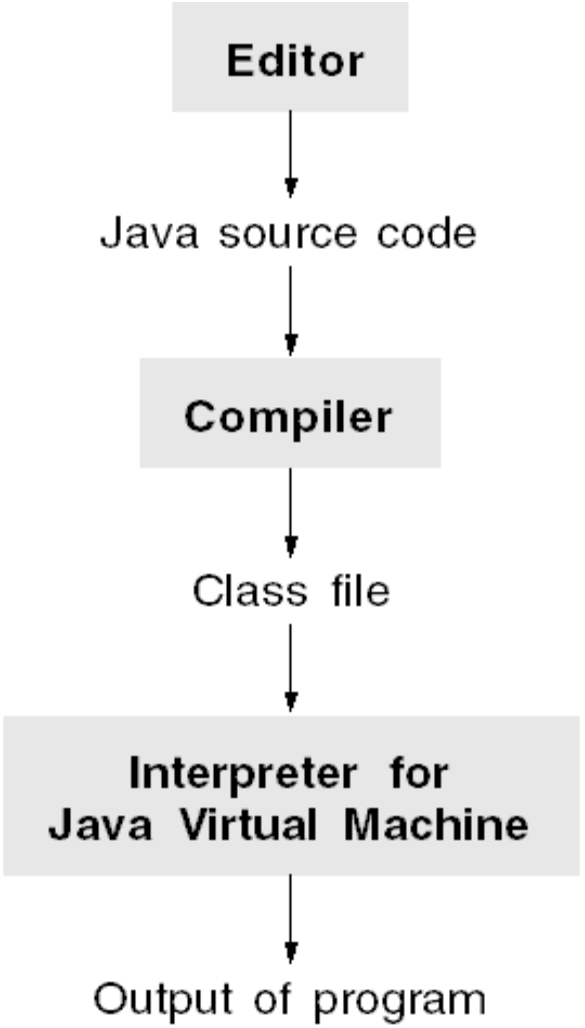
Test the solution

Machine test the code for syntax and logic errors

- Compile the program
  - Syntax errors are detected by the compiler
  - If compile-time errors exist, an executable version of the program is not created
  - Fix any errors until the code compiles
- Execute (run) the program
  - Logic errors are detected by your test cases
  - Run each test case through the program
  - Fix logic errors until test case actual output matches expected output

After running test cases

Input variable name or prompt	Input value	Expected output	Actual output
Enter fahrenheit temperature:	300	148	√
	212	100	√
	100	37	√
	99	37	√
	0	-17	√
	-1	-18	√
	-40	-40	√


Editor
Java source code
Compiler
Class file
Interpreter for
Java Virtual Machine
Output of program
MyProgram.java
(text)
MyProgram.class
(bytecode)

Development Environments

- An ***integrated development environment*** (IDE) is an collection of software tools for developing and testing programs
  - A programmer can write a program, compile it, and execute it, all without leaving the IDE
- There are many IDE's which develop Java software:
  - Borland JBuilder
  - Microsoft Visual J++
  - BlueJ
- Though the details of these environments differ, the basic

Lecture Outcomes

- Today we have covered:
  - the requirements of the unit in terms of assessment and reference material
  - the seriousness of academic misconduct
  - the importance of high quality software
  - programming languages
  - the six stages in problem solving

- Questions?