CITY UNIVERSITY LONDON

ME1107 Computing 1

2008-2009

Y Yan

http://www.staff.city.ac.uk/~ensyy

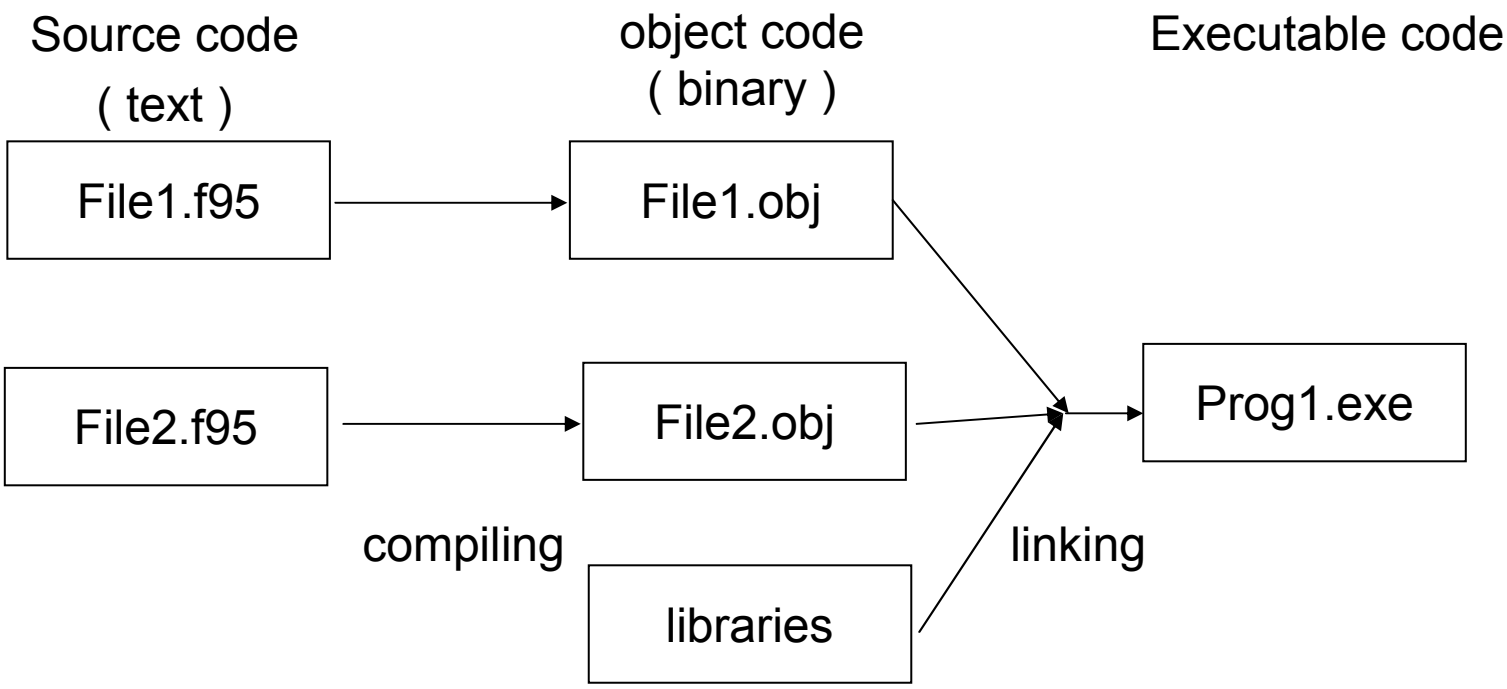
CITY UNIVERSITY LONDON

About Fortran

- Fortran – Formula Translation
- High level computer language
  - Basic, Fortran, C, C++, Java, C#,……(Matlab)
- What do we learn?
  - basic programming skills
  - self-learning skills
- References
  - Text book (Fortran 77 for Engineers)
  - Fortran95 Manual (course website)

CITY UNIVERSITY LONDON

Creating Executable Code



- Source code: text file you write
- Executable code: binary code

CITY UNIVERSITY LONDON

Fortran Programming Environment

Silverfrost Plato, Integrated Development Environment (IDE), is used for Fortran programming

- Create a folder for Fortran programs
- Start Fortran IDE

  Start -> programs -> F. Departmental software -> B. Engineering & mathematics -> Plato IDE

Open a new file:

Free Format Fortran File

CITY UNIVERSITY LONDON

Exercise 1: Your first Fortran program

```
!
!   Your name, Group
!   Date
!
! Hello World

Program Prog1
Print *,'Hello World'
Print *,' your name, group'
Print *,' email address'
Print *, 'Personal tutor's name'
End Prog1
```

Compile: to create .obj
Build : to create .exe
Start: to run .exe

DOS window – Program output

CITY UNIVERSITY LONDON

Data types

In Fortran there are 5 *intrinsic* (i.e. built-in) data *types*:

- Integer: whole numbers
  - 1 bit – 0/1
  - 1 byte – 8 bits
  - For n bits, the integer number range is $(-2^{n-1}, 2^{n-1}-1)$.
  - 2-byte integer: -32768 to 32767
- Real: numbers have a decimal point
  - Finite machine precision
  - Real – 4 byte (seven significant digits and be within the magnitude range of 10^{-38} to 10^{+38})
  - Double – 8 byte
- Complex: real and imaginary parts
- Character: strings of characters
  - 1 character – 1 byte
  - ASCII characters– All letters and symbols on the keyboard mapped with numbers 0-127 (e.g. a – 97, A – 65)
- Logical: .TRUE. or .FALSE.

The first three are the numeric types. The last two are non-numeric types.

CITY UNIVERSITY LONDON

Variable types

- Declaration of variables are not compulsory in the Fortran language
- Without declarations
  - I, J, K, L, M, N, O are integer type
    (Don't use O, it will be confused with 0.)
  - the rest are real type
- Declaration of variables are compulsory in coursework
- IMPLICITY NONE (must be included in program)
  - which means that all variables must be declared

CITY UNIVERSITY LONDON

Declaration of Variables

- Integer

```
integer i, j, k
integer :: n = 0          ! initialised
```

- Real

```
real a, b, c
real x, y, z
real :: sum = 0           ! initialised
```

- Complex
- Char
- logical

FORTRAN is not case sensitive. (e.g. I is same as i)

CITY UNIVERSITY LONDON

Output statement

The syntax of output statement:

WRITE (output_device, format[, options]) list

PRINT format, list (standard output device, usually the screen)

Print *,'Hello World'

Print *,' your name'

Print *,'Input A, B, C'

Print *, 'Roots are ', ROOT1, ROOT2

Write(*,*) 'Roots are ', ROOT1, ROOT2

Text is put between " " or ' '

Variable in the write list: values in variables are printed (not variable names).

CITY UNIVERSITY LONDON

Input statement

The syntax of input statement:

READ (input_device, format[, options]) list_of_variables

READ format, list_of_variables (standard input device, usually the keyboard)

```
Print *,'Input A, B, C'    ! A prompt before input
Read *, A, B, C

Read(*,*) A, B, C
```

CITY UNIVERSITY LONDON

Operators and intrinsic functions

- Numerical operators: +, -, *, /, **

- Logical operators:

  > (.gt.) >= (.ge.) < (.lt.) <= (.le.)

  == (.eq.) /= (.ne.)

- Intrinsic functions:

  Full list is on page 35 of Fortran95 Manual

Some examples:

COS(X), SIN(X), TAN(X) – trigonometric functions (arguments are in *radians)*

EXP(X), LOG(X), LOG10(X) – exponential and logarithmic functions

SQRT(X) – square root

ABS(X) – absolute value (integer, real or complex)

CITY UNIVERSITY LONDON

Type Conversion

- When a binary operator has operands of different type, the weaker (usually integer) type is converted to the stronger (usually real) type and the result is of the stronger type. e.g.

  3 / 10.0 to 3.0 /10.0 = 0.3

- If an integer is divided by an integer then the result must be an integer and is obtained by truncation towards zero.

  3 / 10 = 0

If you intend a constant to be a real number, *use a decimal point!*

CITY UNIVERSITY LONDON

Line discipline

- The usual layout of statements is one-per-line, interspersed with blank lines for clarity.

- There may be more than one statement per line, separated by a *semicolon; e.g.*

  A = 1; B = 10; C = 100

  This is only recommended for simple initialisation.

- Each statement may run onto one or more continuation lines if there is an ampersand (&) at the end of the line to be continued. e.g.

```
DEGREES = RADIANS * PI &
                    / 180.0
```

CITY UNIVERSITY LONDON

Exercise 2: Run the quadratic equation solver

3. A SIMPLE PROGRAM

Example. Quadratic equation solver (real roots).

The well-known solutions of the quadratic equation

$$Ax^2 + Bx + C = 0$$

are

$$x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

The roots are real if and only if the *discriminant* $B^2 - 4AC$ is greater than or equal to zero.

A program which asks for the coefficients and then outputs the real roots might look like the following.

```
PROGRAM ROOTS
! Program solves the quadratic equation Ax**2+Bx+C=0
   IMPLICIT NONE

   REAL A, B, C                                  ! declare variables
   REAL DISCRIMINANT, ROOT1, ROOT2


   PRINT *, 'Input A, B, C'                      ! request coefficients
   READ *, A, B, C

   DISCRIMINANT = B ** 2 - 4.0 * A * C           ! calculate discriminant

   IF ( DISCRIMINANT < 0.0 ) THEN
      PRINT *, 'No real roots'
   ELSE
      ! Calculate roots
      ROOT1 = ( -B + SQRT( DISCRIMINANT ) ) / ( 2.0 * A )
      ROOT2 = ( -B - SQRT( DISCRIMINANT ) ) / ( 2.0 * A )
      PRINT *, 'Roots are ', ROOT1, ROOT2        ! output roots
   END IF

END PROGRAM ROOTS
```

This example illustrates many of the features of Fortran.

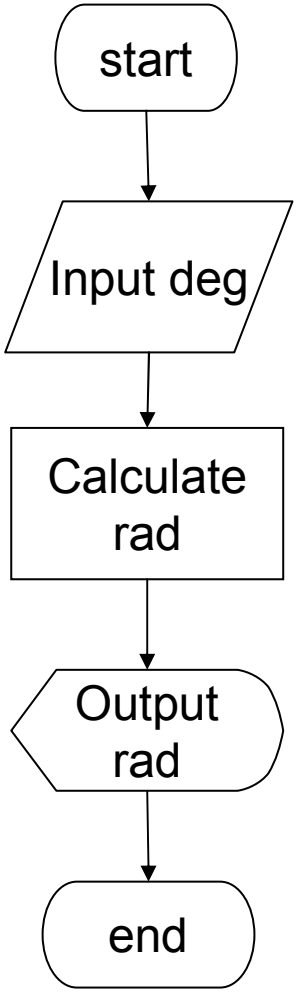
CITY UNIVERSITY LONDON

Exercise 2: Run the quadratic equation solver

- Program structure
- Comment lines
- Implicit None
- Declare variables
- Output: print
- Input: read
- Calculations: =
  one variable on the left side
  variables on the right side much have values
- IF statement

CITY UNIVERSITY LONDON

Exercise 3. Write a program for this flowchart



deg and rad are two variables, which are the names given to two memory locations for storing data.