

SELECTION CONTROL STRUCTURES

Study chapter 5 thoroughly from the text and also study the course notes to learn about selection control structures. Fortunately, ELF90 compiler supports most of the materials that you can find in chapter 5 of the textbook. Therefore, it is important and worthwhile to study chapter 5 of the textbook thoroughly to learn about relational operators, logical operators, IF statements, and CASE statements. You should find the modified example programs on the web. Those programs were taken from the book and then modified to compile and run in ELF90. You should download them, understand them, type them, and test them to gain confidence on the materials.

Cautions in Using Relational and Logical Operators:

1. Do not confuse the equality operator (==) with the assignment operator (=).
2. An expression like $(i < j < k)$ is not legal in Fortran; instead use the form $(i < j .AND. j < k)$.
3. Do not use the equality (==) operator or the not equality operator (/=) to compare to floating point numbers

Practice Problems:

1. **Experiment and Learn:** To build your confidence on the relational operators and the logical operators, you should type and run the following program. Before you run your program, walk through the program and find out the output for each WRITE statement and then run the program to check and verify your output with the computer output. In case of any mismatch, you should check your knowledge on the relevant materials (or you may need to check the program for errors, if any). Each WRITE statement will give output either T (meaning .TRUE.) or F (meaning .FALSE.). Although only one set of values of the variables i, j, and k are used in the program, you should try with many different sets of values for them and see what happens to the output.

```
PROGRAM test
IMPLICIT NONE
INTEGER:: i = 5, j = 3, k = -5
LOGICAL:: a, b

a = (i == j)
b = ( i >= j)
WRITE(*,*) a
WRITE (*,*) b

WRITE (*,*) (2*3 == 6)
WRITE (*,*) (1+4 < 4)
WRITE (*,*) (i*2+4 >= 2*j+3)
WRITE (*,*) (i*3+2 <= 2*j .AND. k+10 > 4)
WRITE (*,*) (2*i+1 >= 2*j .OR. -k+2 >4)
WRITE (*,*) (3>2 .OR. 1+2<3 .AND. 4<=3)
WRITE (*,*) (.NOT.(i+j > -k+j))
WRITE (*,*) (i*j+10 <= 20.AND.i > 3 .OR..NOT. i+k < 3)
WRITE (*,*) (-i*k/3 > 4 .OR. 3 < j .AND. .NOT.j/2 == 1)

STOP
END PROGRAM test
```

2. **Coding:** It is important to learn how to code selection structures in Fortran. Here are some simple coding problems that you should try to code in Fortran. (The solution for one of them is given as an example.)

- a. Write Fortran statements to increment an integer variable a_count by one if the test score (test_score) of a student is 90 or over.
- b. Write Fortran statements to increment an integer variable b_count by one if the test score (test_score) of a student is in the range of 80 to 89.
- c. Write an if statement to check whether the value of an integer variable count is divisible by 3 or not. Accordingly, print the message "It is divisible by 3" or "It is not divisible by 3".
- d. Write a Fortran statement to check whether a real variable is positive or not. If it is positive, print the square root of the variable; otherwise print "The number is not positive".
- e. Write Fortran statements to check whether a real variable is positive, zero, or negative. If it is positive, print the square root of the number; if it is zero, print "The square root of the number is zero"; if it is negative, change the sign of the variable (i.e., variable = - variable) and then calculate and print the square root of the number with a proper message.
- f. Write if-else statements to calculate the sales commission of a salesperson as follows:
sales commission = 3% of the sales amount if sales amount is over \$50000 or the commission code is 1; otherwise, sales commission = 2% of the sales amount.

- g. Write if-else statements to calculate the tuition fee of a student as follows:

| <u>Credit Hours</u> | <u>Tuition Fee</u> |
|---------------------|-----------------------------------------------------|
| Not over 15 hours | \$200 per credit hour |
| Over 15 hours | \$3000 plus \$150 per credit hour for hours over 15 |

solution:

```

IF (credit_hours <= 15) THEN
    tuition_fee = credit_hours * 200.0
ELSE
    tuition_fee = 3000.0 + (credit_hours - 15) * 150.0
END IF

```

- h. Suppose in one state, residents are subject to the following tax:

| <u>Income</u> | <u>Amount of Tax</u> |
|-----------------------|-------------------------------------|
| Not over \$1000 | 1% of income |
| Over \$1000 to \$3000 | \$10 plus 2% of amount over \$1000 |
| Over \$3000 to \$7000 | \$50 plus 3% of amount over \$3000 |
| Over \$7000 | \$170 plus 4% of amount over \$7000 |

Write a program segment using IF..ELSE..END IF statements to calculate the tax for a resident of the state.

Final Note: The next step of learning "selection structures" is to understand and try the example programs handed out in the class and the example programs found in the text book. Although, project 3 will give you some opportunity to exercise your knowledge on "selection structures", for those who do not have any prior programming experience, I would suggest you try Programming Exercises 5.10 and 5.11 (page 165) from the textbook for more exercise.