Practical No. 3 Date: / / 201

Aim: Program to display student result based on given subjects (at least 3).

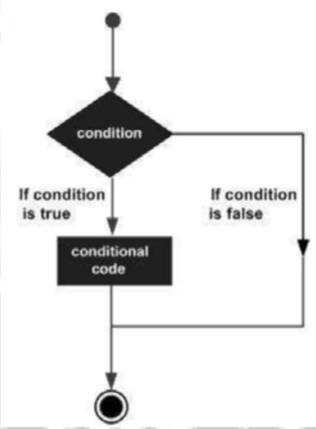
Objectives:

- To study R conditions.
- Implement a program to display student result.

Theory:

R Conditions

Decision making structures require the programmer to specify one or more conditions to be evaluated. After evaluation a statement or statements will be executed if the condition is determined to be true. Optionally, other statements to be executed if the condition is determined to be false.



R provides the following types of decision making statements.

Statement	Description
if	An if statement consists of a Boolean expression followed by one or more statements
if...else	An if statement can be followed by an optional else statement, which executes when the Boolean expression is false.
if...else if...else	An if statement can be followed by an optional else if statement, allows multiple Boolean expression to be tested, followed by an optional else which executes when the Boolean expression is false.
switch	A switch statement allows a variable to be tested for equality against a list of values.

if Statement

Syntax

```
if(boolean_expression){
// statement(s) will execute if the boolean expression is true.
}
```

Example

```
X <- 30L
if(is.integer(x)){
     print(“X is an integer”)
}
```

Output

```
[1] “X is an integer”
```

if..else Statement

Syntax

```
if(boolean_expression){
// statement(s) will execute if the boolean expression is true.
} else{
// statement(s) will execute if the boolean expression is false.
}
```

Example

```
X <- c(“what”, “is”, “truth”)
if(“Truth” %in% X){
     print(“Truth is Found”)
} else{
     print(“Truth is Not Found”)
}
```

Output

```
[1] “Truth is Not Found”
```

if..else if..else Statement

Syntax

```
if(boolean_expression#1){
// statement(s) will execute if the boolean expression #1 is true.
} else if(boolean_expression#2){
// statement(s) will execute if the boolean expression #2 is true.
} else{
// statement(s) will execute if no boolean expression is true.
}
```

Example

```
X <- c(“what”, “is”, “truth”)
if(“Truth” %in% X){
     print(“Truth is Found”)
} else if(“truth” %in% X){
     print(“truth is Found”)
} else{
{
     print(“Neither truth or Truth is Found”)
}
```

Output

```
[1] “truth is Found”
```

switch Statement

Following are rules regarding use of switch statements:

- If the value of expression is not a character string it is coerced to integer.
- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
- For integers expression evaluated for 1 to max number of integer arguments.
- For a character string the expression exactly match string to the names of the elements.
- If there is more than one match, the first matching element is returned.
- No Default argument is available.
- In the case of no match, if there is a unnamed element & its value is returned. (If there is more than one such argument an error is returned.)

Syntax

```
switch(expression, case#1, case#2, case#3, …)
// statement(s) to be executed based on expression evaluation.
```

Example

```
X <- switch(
     3,
     “One”,
     “Two”,
     “Three”,
     “Four”
)
```

Example

```
[1] “Three”
```

Algorithm

1. Start.
2. Read marks obtained for each subject (at least 3).
3. Calculate total of marks obtained.
4. Display total marks with out of marks.
5. Calculate average percentage from total marks.
6. Based on percentage display appropriate result as,
    a. If percentage is greater than 75, display “First Class with Distinction/ A grade”
    b. Else-If percentage is greater than 60, display “First Class / B grade”
    c. Else-If percentage is greater than 50, display “Second Class / C Grade”
    d. Else-If percentage is greater than 40, display “Pass / D grade”
    e. Else display “Fail / F grade”
7. Stop.