Finolex Academy of Management and Technology, Ratnagiri

Department of Information Technology

Subject:	**R Programming Lab. (ITL804)**		
Class:	**BE IT / Semester – VIII (Rev-2016) / Academic year: 2019-20**		
Name of Student:	**Kazi Jawwad A Rahim**		
Roll No:	**28**	**Date of performance (DOP) :**	31/01/2020
Assignment/Experiment No:	**03**	**Date of checking (DOC) :**	19/03/2020
Title: Program to demonstrate flow control instructions and functions			
Marks:	08	**Teacher's Signature:**	

1. Aim: To understand the use of various flow control instructions and functions in R.

2. Prerequisites:

1. Basics of R programming, various data structures used in R etc.

3. Hardware Requirements:

1. PC with minimum 2GB RAM

4. Software Requirements:

1. Windows / Linux OS.
2. R version 3.6 or higher

5. Learning Objectives:

1. To understand decision and loop control instructions.
2. To understand function definition and calling to it.

6. Learning Objectives Applicable: LO 1

7. Program Outcomes Applicable: PO 1, PO 2

8. Program Education Objectives Applicable: PEO 2

Theory:
There are 8 types of control structures in R:
i) If
ii) If-else
iii) For
iv) Nested loops
v) while
vi) Repeat and Break
vii) Next
viii) Return

i) If - This task is carried only if the condition is returned as TRUE.

Syntax -
```
if(expression){
    statements
}
```

2) If-else - An if-else statement contains the same elements as an if statement, with some extra elements.

Syntax -
```
if(expression){
    ----
}else{
    ----
}
```

3) For - A loop is a sequence of instructions that is repeated until a certain condition is reached.

Syntax -
```
for( id in 1:5){
    print(id)
}
```

4) while - The format is while(condition) expr, where condition is the condition to test and expr is an expression. for e.g.,

```
val = 2.87
while( val <= 4.87){
    val = val + 0.87
    print( c(val, val-2, val-1))
}
```

5) Repeat & Break -

- we use break statements inside a loop to stop the iterations and flow control outside of the loop. while in a nested looping situation, where there is a loop inside another loop, this statement exists from the innermost loop that is being evaluated.

- A repeat loop is used to iterate over a block of code, multiple number of times. There is no condition check in a repeat loop to exit the loop. We ourselves put a condition explicitly inside the body of the loop and use the break statement to exit the loop. Failing to do so will result in infinite loop.

Syntax:

```
repeat{
    ~~# simulations~~
    if (value - expectations) <= threshold){
        break
    }
}
```

6) Next statement in R

Next jumps to the next cycle without completing a particular iterations. In fact it jumps to the evaluation of the condition holding the current loop. Next statement enables to skip the current iteration of a loop without terminating it.

for e.g.;

```
x = 1:4
for(i in x){
    if(i==2){
        next
    }
    print(i)
}
```

O/P: 1, 3, 4

7) Return statement in R -

Many times, we will require some functions to do processing and return back the result. This is accomplished with the return() statement in R.

Syntax: return(expression)

OUTPUT:

IF ELSE Example:

```
age=as.numeric(readline("Enter age"))
gender=readline("Enter Gender")
if(age>=60 && gender=="M"){
 print("Available for Concession")
}else if(age>=45 && gender=="F"){
 print("Available for Concession");
}else{
 print("Noot avaialable for Concession")
}
```

OUTPUT:

```
> source('D:/JK/If Else.R')
Enter age60
Enter GenderM
[1] "Available for Concession"
```

SWITCH:

```
day=as.numeric(readline("Enter Day Number\n"))
y=switch(day,"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday")
print(y)
```

OUTPUT:

```
> source('D:/JK/Switch.R')
Enter Day Number
5
[1] "Friday"
```

For:

```
for(i in 1:10){
 print(i)
}
```

OUTPUT:

```
> source('D:/JK/For.R')
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

While:

```
i=1
while(i<=5){
 print(i)
 i=i+1
}
```

OUTPUT:

```
> source('D:/JK/While.R')
[1] 1
[1] 2
```

```
[1] 3
[1] 4
[1] 5
```

Repeat:

```
i=1
repeat{
  print(i)
  i=i+1
  if(i>5){
    break
  }
}
```

OUTPUT:

```
> source('D:/JK/Repeat.R')
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Function:

```
area = function(l,w){
  a=l*w
  return(a)
}
print(area(3,5))
```

OUTPUT:

```
> source('D:/JK/Function.R')
[1] 15
```

Double Function:

```
volume=function(r,l){
  area=function(r){
    a=r*r
    return(a)
  }
  v=area(l)*3.14*r*l
  return(v)
}
print(volume(3,5))
```

OUTPUT:

```
> source('D:/JK/Double Function.R')
[1] 1177.5
```

Learning Outcomes Achieved:

1. We understood decision and loop control instructions.
2. We understood function definition and calling to it.

Conclusion:

We have successfully demonstrated the loop instructions like If-Else, Switch, For, While, Repeat and functions and double functions.

13. Experiment/Assignment Evaluation

Experiment/Assignment Evaluation:				
Sr. No.	**Parameters**		**Marks obtained**	**Out of**
1	Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used -		05	6
2	Neatness/presentation		02	2
3	Punctuality		01	2
Date of performance (DOP)	21/01/2020	**Total marks obtained**	08	**10**
Date of checking (DOC)	19/03/2020	**Signature of teacher**		

References:

1. URL: https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf (Online Resources)
2. R Cookbook Paperback – 2011 by Teetor Paul O Reilly Publications
3. Beginning R: The Statistical Programming Language by Dr. Mark Gardener, Wiley Publications
4. R Programming For Dummies by Joris Meys Andrie de Vries, Wiley Publications

Viva Questions

1. What are decision control instructions ?
2. What are loop control instructions ?
3. Compare flow control instructions in R with flow control instructions in Python ?
4. How to define function in R?
5. Can I shuffle arguments of the functions while calling it?