Assignment–01 : R Programming (BCA–6th Semester)

Subject Code: BCA-601
Student: Suraj Kumar
University: M.M. (Deemed to be University), Mullana
Session: 2025–26

Q1. Write a simple R program to print "Hello, World!" to the console. (1 Mark)

In R programming, the `print()` function is used to display output on the console.
This function shows messages or values on the screen.
"Hello, World!" is usually the first program written by beginners to understand basic syntax.

```
print("Hello, World!")
```

Q2. In what areas is R primarily used, and why is it popular in data science? (1 Mark)

R is mainly used in data analysis, statistics, and research work.
It is widely used in business analysis, banking, healthcare, and machine learning.
R is popular in data science because it provides strong analytical and visualization tools.

Q3. How would you assign a value to a variable in R? Write the syntax using both `<-` and `=` operators. (1 Mark)

In R, a variable is used to store data values.
Values can be assigned to variables using `<-` or `=` operators.
Both operators perform the same task of assignment.

```
x <- 10
y = 20
```

Q4. Describe the difference between the `==` operator and the `=` operator in R. (1 Mark)

The `=` operator is used to assign a value to a variable.
The `==` operator is used to compare two values.
It returns TRUE if values are equal, otherwise FALSE.

Q5. What is the length of x? `x <- 5:10` (1 Mark)

The expression `5:10` creates a sequence from 5 to 10.
It includes six numbers: 5, 6, 7, 8, 9, and 10.
Therefore, the length of vector x is 6.

```
x <- 5:10
length(x)
```

Q6. Create a vector of integers in R. How can you check the type of the vector after its creation? (1 Mark)

A vector in R is created using the `c()` function.
After creating a vector, its data type can be checked using the `typeof()` function.
This helps to identify the kind of values stored in the vector.

```
v <- c(1, 2, 3, 4)
typeof(v)
```

Q7. How would you handle missing data (NA) in R? Provide a method and demonstrate with an example. (2 Marks)

In R, missing values are represented by `NA`.
Missing data may occur due to incomplete information.
The function `is.na()` is used to identify missing values.
To remove missing values, the `na.omit()` function is used.
Handling NA values improves data accuracy.

Example:

```
x <- c(10, 20, NA, 40)
is.na(x)
na.omit(x)
```

This method removes the missing value from the data.

Q8. What is subsetting in R? Provide an example using a vector. (2 Marks)

Subsetting means selecting specific elements from a data object.
In R, subsetting is done using square brackets `[]`.
It allows users to extract required data for analysis.
Subsetting helps in efficient data handling.

Example:

```
x <- c(10, 20, 30, 40)
x[2]
x[1:3]
```

Q9. List and explain the main data types in R. How do they differ from each other? (4 Marks)

R programming supports different data types to store various kinds of data.
Each data type has a specific purpose.

1. Numeric

Numeric data type stores decimal and real numbers.
Examples include 10.5, 25.7, and 3.14.
It is mainly used for calculations.

2. Integer

Integer data type stores whole numbers without decimal points.
Integers are written using the letter `L`.
Example: 10L, 50L.

3. Character

Character data type is used to store text values.
It is written inside quotation marks.
Example: "Data Analysis".

4. Logical

Logical data type stores TRUE or FALSE values.
It is commonly used in decision-making conditions.

Difference

These data types differ based on the type of values they store, such as numbers, text, or logical values.
Using correct data types helps in writing accurate and efficient programs.

Q10. What are the best practices for naming variables in R? Evaluate how these practices contribute to writing clean and maintainable code. (4 Marks)

Variable naming plays an important role in programming.
Good variable names make the code easy to read and understand.

Best Practices:

1. Use meaningful variable names (e.g., `total_marks`).
2. Avoid spaces in variable names.
3. Use lowercase letters for consistency.
4. Use underscore (_) to separate words.
5. Do not start variable names with numbers.

Importance

Proper naming improves code readability.
It reduces confusion while debugging.
Clean variable names make programs easy to maintain.
They help other programmers understand the logic quickly.

Q11. List and describe the key features of the R programming language. Evaluate why these features make R suitable for data science applications. (6 Marks)

R is a powerful programming language mainly developed for data analysis and statistical computing.
It provides many advanced features that support data science work.

1. Open Source

R is free to use and open source.
Anyone can access and modify it.

2. Large Package Collection

R provides thousands of packages such as ggplot2, dplyr, and tidyverse.
These packages simplify data analysis tasks.

3. Strong Statistical Support

R includes built-in statistical functions.
It supports regression, correlation, hypothesis testing, and probability analysis.

4. Excellent Data Visualization

R is well known for high-quality graphs and charts.
Visualization helps in understanding data patterns easily.

5. Efficient Data Handling

R can handle large datasets efficiently.
Data cleaning and transformation are simple in R.

6. Data Science Friendly Language

R supports the complete data science process.
Therefore, it is widely used by data analysts and researchers.

Q12. Explain the use of operators in R and the BODMAS rule. Solve the following equation step by step. (6 Marks)

Operators are symbols used to perform operations on values in R.

Types of Operators in R

1. Arithmetic Operators: +, -, *, /, %%
2. Relational Operators: >, <, ==, !=
3. Logical Operators: &, |, !
4. Assignment Operators: <-, =

BODMAS Rule

R follows the BODMAS rule while evaluating expressions:

B – Brackets
O – Order (power)
D – Division
M – Multiplication
A – Addition
S – Subtraction

Given Expression:

```
5 + 4 * 9 %% (3 + 1) / 6 - 1
```

Step-by-Step Solution:

Step 1: Solve brackets (3 + 1) = 4

Step 2: Modulus operation 9 %% 4 = 1

Step 3: Multiplication 4 * 1 = 4

Step 4: Division 4 / 6 = 0.66

Step 5: Addition 5 + 0.66 = 5.66

Step 6: Subtraction 5.66 - 1 = 4.66

Final Answer:

Result = 4.66 (approximately)

✅**All answers are written in simple English and extended form for examination use.**