2025.03.06

SWT11022: Practical for Fundamentals of Programming

Department of Information & Communication Technology

Faculty of Technology

South Eastern University of Sri Lanka

Time: - 08.30 am – 9.30am Lab Sheet 03

Title: Introduction to the Variables in C Programming

Objectives:

- Understand the basics of variables, including declaration, initialization, assignment, and scope.
- Understand constants and demonstrate their declaration, initialization, and usage.

Practical 1: Working with Variables.

Steps:

1. Declaration of Variables:

Declare three variables of different data types: int, float, and char.

```
int age;

float temperature;

char grade;
```

2. Initialization of Variables:

Initialize these variables with appropriate values.

```
int age = 25;

float temperature = 98.6;

char grade = ‘A’;
```

3. Display Values:

Use printf statements to display the values of these variables.

```
printf(“Age: %d\n”, age);

printf(“Temperature: %f\n”, temperature);

printf(“Grade: %c\n”, grade);
```

4. Variable Assignment:

Change the values of these variables using assignment.

```
age = 30;

temperature = 99.5;

grade = “B”;
```

5. Display Updated Values:

Use printf statements to display the updated values of the variables.

```
printf(“Updated Age: %d\”, age);

printf(“Updated Temperature: %f\n”, temperature);

printf(“Updated Grade: %c\n”, grade);
```

6. Naming the variable

Check whether these variables are used.

```
int Age =23;

int age = 36;

int aGe = 40; int 9age =20;
```

2025.03.06

Practical 2: Working with Constants

1. Declaration of Constants (Using #define):

Declare three constants using #define. These constants can represent mathematical or scientific constants.

```
#define PI 3.14159
#define GRAVITY 9.81
#define SPEED_OF_LIGHT 299792458
```

2. Declaration of Constants (Using const):

Declare three constants using the const keyword.

```
const int MAX_SCORE = 100;
const char APP_NAME[] = “MyApp”;
const float TAX_RATE = 0.08;
```

3. Display Constants:

Use printf statements to display the values of the constants declared with both methods.

```
printf(“The value of PI: %f\n”, PI);
printf(“The value of MAX_SCORE: %d\n”, MAX_SCORE);
```

4. Changing Constants (Using #define):

Constants declared with #define cannot be changed in the program.

```
// Attempting to change a #define constant (generates a compilation error)
// PI = 3.14;
```

5. Changing Constants (Using const):

Const constants are not meant to be changed, but it is possible to declare a variable with the same name within a smaller scope.

```
const int MAX_SCORE = 90; // Creating a new variable within a smaller scope.
```

Tasks

1. A university wants to store basic details of a student in a C program. Write a program that:

    1) Declares three variables:

        - int studentID (to store the student’s ID number)
        - float GPA (to store the student’s Grade Point Average)
        - char grade (to store the student’s current grade)

    2) Initializes them with the values:

        - studentID = 12345
        - GPA = 3.75
        - grade = 'A'

    3) Displays these values using printf ().
    4) Updates the values to studentID = 54321, GPA = 3.90, and grade = 'A+'.
    5) Displays the updated values.

2. A physics student is learning about fundamental scientific constants.
   Write a C program that:

    1) Declares the following constants using `#define`:

        a. `SPEED_OF_LIGHT = 299792458` (Speed of light in m/s)

        b. `GRAVITY = 9.81` (Acceleration due to gravity in m/s^2)

        c. `PLANCK_CONSTANT = 6.626e-34` (Planck's constant in J·s)

    2) Declares the following constants using `const`:

        a. `const float BOLTZMANN_CONSTANT = 1.38e-23;`

        b. `const int ABSOLUTE_ZERO = -273;` (Absolute zero in Celsius)

    3) Prints all constant values using `printf().`

3. A bank software development team is writing a C program to store customer transaction details. The following variable names are suggested:

```
int AccountBalance = 5000;
int account_balance = 4500;
int 2ndTransaction = 200;
int transaction Amount = 150;
int float = 100;
```

    1) Identify which variable names are **valid** and **invalid** based on C programming rules.

2) Explain why invalid names are incorrect.
3) Rewrite the correct variable declarations and print their values using `printf()`.

Report Submission Guidelines

- Submit the Report by 10/03/2025.
- Late submissions will not be accepted.
- Report Structure
    - Practical No
    - Date of Submission
    - Title
    - Objective of the practical.
    - Exercise
    - Challenges
    - Conclusion
    - References