

**SWT11022: Practical for Fundamentals of Programming**  
**Department of Information & Communication Technology**  
**Faculty of Technology**  
**South Eastern University of Sri Lanka**

Academic Year: 2022/2023

Lab Sheet 02

---

Submission Date: 24/02/2025**Title:** Introduction to the Basic Data Types in C Programming**Objectives:**

- Familiarize with integer and floating-point data types and their usage.
- Understand character data types and work with strings.
- Explore data type modifiers and user-defined data types.

---

**Practical 1: Working with Integer and Floating-Point Data Types**

---

**Steps:**

## 1. Create an Integer Variable:

```
int age = 25;
```

## 2. Create a Floating-Point Variable:

```
float pi = 3.14159;
```

## 3. Perform Arithmetic Operations:

- Perform arithmetic operations using the integer and floating-point variables.
- For example, you can calculate the sum, difference, product, etc.

```
float price = 5.99;  
int quantity = 3;  
float total = price * quantity;
```

## 4. Display Results:

- Print to display the results of the arithmetic operations.

```
printf("Total price: %.2f\n", total);
```

## Practical 2: Character Data Type and Strings

---

Steps:

1. Create a Character Variable:

```
char grade = 'A';
```

2. Display the Character:

```
print("Grade: %c\n",grade);
```

1. String Manipulation:

Introduce the concept of strings as character arrays.

Declare a string and initialize it with a phrase.

```
char phrase [] = "Hello, World!";
```

2. String Functions:

- string functions like `strlen()`, `strcpy()`, and `strcat()`.

```
int length = strlen(phrase);
```

```
char destination[50];
```

```
strcpy(destination, phrase);
```

3. Display Results:

```
printf("String Length: %d\n", length);
```

```
printf("Copied String: %s\n", destination);
```

## Practical 3: Modifiers and Custom Data Types

---

Steps:

### 1. Using Data Type Modifiers:

- Introduce data type modifiers like short, long, signed, and unsigned.
- declare variables using these modifiers.

```
short distance = 1000;
```

```
long population = 70000000000;
```

```
unsigned int score = 95;
```

### 2. Display Results:

- Use printf to display the variables and values and data type sizes.

```
printf("Distance: %d\n", distance);
```

```
printf("Population: %ld\n", population);
```

```
printf("Score: %u\n", score);
```

### 3. Creating a User-Defined Data Type:

- Introduce the concept of structures.
- define a structure representing a book with attributes like title, author, and price.

```
struct Book {
```

```
char title[100];
```

```
char author[50];
```

```
double price;
```

```
};
```

### 4. Declare and Initialize:

- declare and initialize variables of the user-defined data type.

```
struct Book myBook;
```

```
strcpy(myBook.title, "The Catcher in the Rye");
```

```
strcpy(myBook.author, "J.D. Salinger");
```

```
myBook.price = 12.99;
```

## 5. Display Book Information:

- printf to display the book's information.

```
printf("Book Title: %s\n", myBook.title);
```

```
printf("Author:  %s\n",myBook.author);
```

```
printf("Price: $%.2f\n",myBook.price);
```

## Tasks:

1. Write the following code to understand:
  - a. the different data types
  - b. variable declarations & initialization.
  - c. Getting output using printf()

```
int id = 101;
char section = 'B';
float marks = 89.5;
printf("ID: %d, Section: %c, Marks: %.2f\n", id, section, marks)
```

2. Write a program that asks the user for total worked hours and pay rate per hour. Then, calculate and display the total salary. (using the scanf())
3. Execute the code below.
  - a. Identify and correct errors in this code.
  - b. What are the keywords used in this code?
  - c. What are the escape sequences used in this code? Define them.

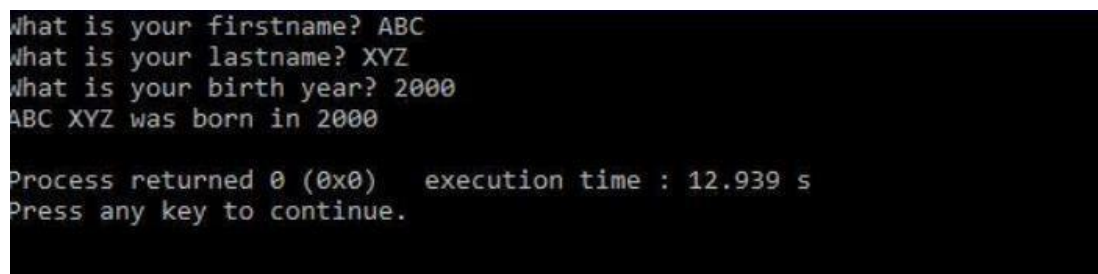
```
#include <stdio.h> int main(){
//variable declaration char name[20]; int a, b; float c, f;
//variable initialization char val[] = "Welcome"; a= 10;
//Defining Constants const float x = 3.73
//Inteaction with the user printf("Enter your name: "); scanf("%s", &name); printf(val);

printf("\nYour name is %d.", name); c= 90.0*3.0
printf("\nValue of c: %f", c); x= 49;
printf("\nValue of X: %f", x);

//using the escape sequences printf("Hello\tWorld\n\n");
printf("You\nare\nlearning\n%c\ language\n\"Do you know C language\");

return 0;
}
```

4. Write a program to provide the following output.



```
What is your firstname? ABC
What is your lastname? XYZ
What is your birth year? 2000
ABC XYZ was born in 2000

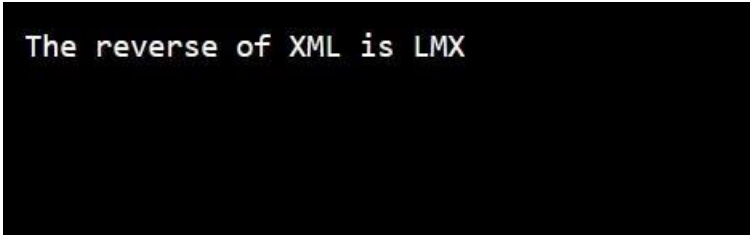
Process returned 0 (0x0)   execution time : 12.939 s
Press any key to continue.
```

5. Write a program to find the area of a rectangle where the length and width of the rectangle is 25.12428 cm and 12.59 cm, respectively. The area should be in four decimal point

**area = length \* width**

6. Write a C program to print the following characters in a reverse way. Only use **printf()** command. Test Characters: 'X', 'M', 'L'

Output:



```
The reverse of XML is LMX
```

### Report Submission Guidelines

- Submit the **Report** by **24/02/2025**.
- Late submissions will not be accepted.

### Report Structure

- Practical No
- Date of Submission
- Title
- Objective of the practical.
- Exercise
- Challenges
- Conclusion
- References