

**CC-211L**

**Object Oriented Programming**

**Laboratory 08**

**Stream I/O and Operator overloading**

**Version: 1.0.0**

**Release Date: 21-03-2024**

**Department of Information Technology**

**University of the Punjab**

**Lahore, Pakistan**

## Contents:

- Learning Objectives
- Required Resources
- General Instructions
- Background and Overview
- C++ Stream
- Type Conversion
  - Activities
- Pre-Lab Activity
  - Streams
  - iostream Library Headers
  - Standard Stream Objects cin, cout
  - Task 01
  - Task 02
  - Task 03
- In-Lab Activity
  - Stream Output
  - Stream Input
  - Scan and display a single character
  - Display ncount
  - istream\_withassign class
  - ostream\_withassign class
  - Overloading Cast Operator
- Submissions
- Evaluations Metric
- References and Additional Material
- Lab Time and Activity Simulation Log

## Learning Objectives:

- Streams
- Stream Input
- Stream Output
- Overloading Cast Operator

## Resources Required:

- Desktop Computer or Laptop
- Microsoft ® Visual Studio 2022

## General Instructions:

- In this Lab, you are **NOT** allowed to discuss your solution with your colleagues, even not allowed to ask how is s/he doing, this may result in negative marking. You can **ONLY** discuss with your Teaching Assistants (TAs) or Lab Instructor.
- Your TAs will be available in the Lab for your help. Alternatively, you can send your queries via email to one of the followings.

Teachers:		
Course Instructor	Prof. Dr. Syed Waqar ul Qounain	<a href="mailto:swjaffry@pucit.edu.pk">swjaffry@pucit.edu.pk</a>
Lab Instructor	Dr.Prof. Sanam Ahmad	
Teacher Assistants	Syed M.Jafar Ali Naqvi	<a href="mailto:bitf21m032@pucit.edu.pk">bitf21m032@pucit.edu.pk</a>
	M. Usman Munir	<a href="mailto:bitf19m020@pucit.edu.pk">bitf19m020@pucit.edu.pk</a>
	Aqdas Shabir	<a href="mailto:bitf21m022@pucit.edu.pk">bitf21m022@pucit.edu.pk</a>
	Ibad Nazir	<a href="mailto:bitf21m024@pucit.edu.pk">bitf21m024@pucit.edu.pk</a>
	Umer Farooq	<a href="mailto:bitf21m010@pucit.edu.pk">bitf21m010@pucit.edu.pk</a>

## **Background and Overview:**

### **C++ Stream:**

Stream in C++ means a stream of characters that gets transferred between the program thread and input or output. There are a number of C++ stream classes eligible and defined which is related to the files and streams for providing input-output operations. All the classes and structures maintaining the file and folders with hierarchies are defined within the file with `iostream.h` standard library. Classes associated with the C++ stream include `ios` class, `istream` class, and `ostream` class.

### **Type Conversion:**

Type conversion is the process that converts the predefined data type of one variable into an appropriate data type. The main idea behind type conversion is to convert two different data type variables into a single data type to solve mathematical and logical expressions easily without any data loss.

## Activities:

### Pre-Lab Activities:

#### Streams:

C++ I/O occurs in streams, which are sequences of bytes. In input operations, the bytes flow from a device to main memory. In output operations, bytes flow from main memory to a device.

#### **iostream Library Headers:**

The C++ stream libraries provide hundreds of I/O capabilities. Most of our C++ programs include the header, which declares basic services required for all stream I/O operations. The header defines the `cin`, `cout`, `cerr` and `clog` objects, which correspond to the standard input stream, the standard output stream, the unbuffered standard error stream, and the buffered standard error stream. Both unformatted and formatted-I/O services are provided.

The `iostream` library provides many class templates for performing common I/O operations.

- **istream** for stream input operations
- **ostream** for stream output operations
- **iostream** for both stream input and output operations

#### **Standard Stream Objects `cin`, `cout`:**

Predefined object `cin` is an `istream` object and is said to be “connected to” (or attached to) the standard input device, which usually is the keyboard. The stream extraction operator (`>>`) as used in

#### **`int marks;`**

#### **`cin >> marks;`**

causes a value for `int` variable `marks` to be input from `cin` to memory. The compiler selects the appropriate overloaded stream extraction operator, based on the type of the variable `marks`. The `>>` operator is overloaded to input data items of fundamental types, strings, and pointer values.

The predefined object `cout` is an `ostream` object and is said to be “connected to” the standard output device, which usually is the display screen. The stream insertion operator (`<<`), as used in the following statement, causes the value of variable `marks` to be output from memory to standard output device:

#### **`cout << marks;`**

**Task 01: Student Marks****[Estimated time 15 minutes / 10 marks]**

- Create a class "Student" with the following data members:
  - Subjects (int)
  - Marks (int [])
- Implement a copy constructor for the "Student" class.
- Implement getter/setter methods
- Implement a method 'inputMarks' which takes no of subjects and marks as input from user
- Implement a method 'outputMarks' which displays the marks of student.

**Task 02: Person Class****[Estimated time 20 minutes / 20 marks]**

Create a class Person with the following private member variables:

- name (string): the name of the person
- age (int): the age of the person
- gender (char): the gender of the person (either 'M' for male or 'F' for female)

Implement the following public member functions:

- A default constructor that sets the name to an empty string, the age to 0, and the gender to 'M'.
- A parameterized constructor that sets the name, age, and gender to the values passed as arguments.
- Write a function to read in a Person object from the standard input. The input should consist of three values: the name (a string), the age (an int), and the gender (a char). The input values should be separated by whitespace. For example, if the input is "John 25 M", the Person object should have name "John", age 25, and gender 'M'.
- Write a function to write a Person object to the standard output in the following format: "Name: [name], Age: [age], Gender: [gender]". For example, if a Person object has name "John", age 25, and gender 'M', the output should be "Name: John, Age: 25, Gender: M".

Once you have implemented the Person class, write a main function that demonstrates how to use the implemented functions.

**Task 03: Output Mania:****[Estimated time 20 minutes / 10 marks]**

Create a program which will keep on taking sentence input from user and display the sentence when user hits Enter key. Then take the next sentence as input and so on until user inputs End of file character. Take input in a int variable character and also display end of file character value from cout before and after the program ends.