Laboratory Manual

For

Programming for Problem Solving-II

(ESC201)

B.Tech (IT)

SEM II

March 2022

Faculty of Technology
Dharmsinh Desai University
Nadiad.
www.ddu.ac.in

Table of Contents

LABWORK BEYOND CURRICULA

COMMON PROCEDURE

Tools / Apparatus: Unix/Linux Operating System, Text Editor, g++ compiler

Procedure:

- Write the code of the program
- Compile the program for any compile-time errors
- Run the program
- Debug the program for any errors

Sample Experiment

1 AIM: Create a class time with data as hours and minutes.

a. Write a function gettime() and puttime() to read and display data of time class.
b. Design a function called sum() that accepts two objects as argument and display the sum of two time objects.

2 TOOLS/APPARATUS: g++ compiler

3 STANDARD PROCEDURES:

3.1 Analyzing the Problem:

- First create a file named as “ti.cpp”.
- After that includes the standard input/output files.
- Now define the class “Time” and its members.
- Create the functions which are necessary for the program.
- Now create the main function and take the information.

3.2 Designing the Solution:

- Create a c++ file named as “ti.cpp”.
- Define the class named as “Time”. Also define its members that are hours and minutes.
- Make functions gettime() and puttime() to set and display time respectively.
- Now in the main function declare 3 Time class objects.
- Now set the data members of objects using gettime() function.
- Finally call sum function which take time objects as argument and store result in calling object.

3.3 Implementing the Solution:

3.3.1 Writing Source Code:

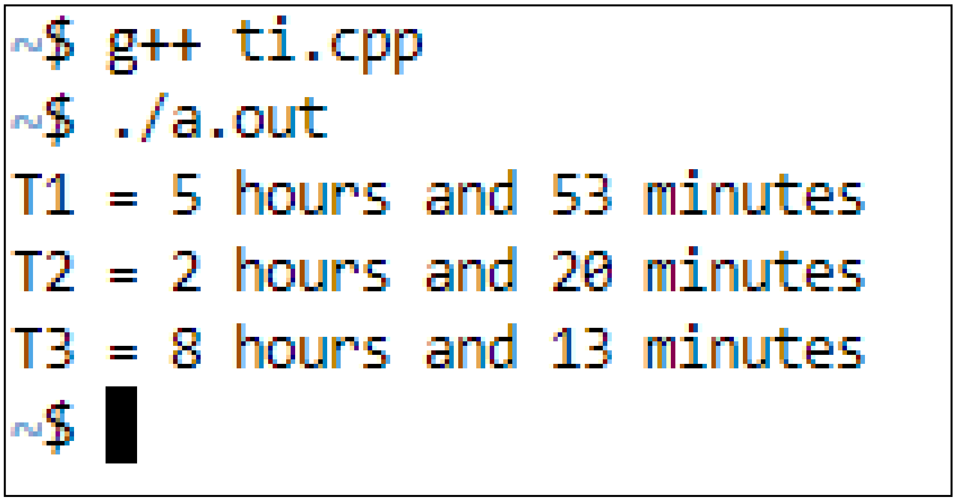
```
#include<iostream>

using namespace std;

class Time
{
   private :
      int hours;
      int minutes;

   public :

      void gettime(int h,int m)
      {
         hours=h;
         minutes=m;
      }
      void puttime(void)
      {
         cout<< hours <<" hours and ";
         cout<< minutes <<" minutes "<<"\n";
      }
      void sum (Time,Time);
};

void Time :: sum(Time t1, Time t2)
{
   minutes = t1.minutes + t2.minutes;
   hours = minutes/60;
   minutes = minutes%60;
   hours = hours + t1.hours + t2.hours;
}

int main()
{
   Time T1, T2, T3;

   T1.gettime(5,53);
   T2.gettime(2,20);

   T3.sum(T1,T2);
   cout<< "T1 = ";
   T1.puttime();
   cout<< "T2 = ";
   T2.puttime();
   cout<< "T3 = ";
   T3.puttime();

   return 0;
}
```

3.3.2 Compilation/Running and Debugging the Solution:

- Compile program by using g++ command

  g++ ti.cpp

- If Successful Compilation is done then Run the Code Using
  ./a.out



```
~$ g++ ti.cpp
~$ ./a.out
T1 = 5 hours and 53 minutes
T2 = 2 hours and 20 minutes
T3 = 8 hours and 13 minutes
~$ █
```

4 Conclusions:

Hence, we have concluded that this experiment will give us the knowledge that how to code a meaningful and understandable program, as well as how to analyze, design and test the program.

Required Software/ Software Tool:

- Linux Operating System
- Terminal (g++ compiler)

Common procedure:

Step 1**:** For the given problem statement design Flowchart/Algorithm/Logic.
Step 2: Define class, variables and functions which will show the flow of program.
Step 3: Write C++ code in the file with .cpp extension.
Step 4: Compile code using g++ compiler, which will create a.out executable file.
Step 5: Test the program using sample input and write down output.

EXPERIMENT-1

Aim: Overview of C++ and basic program of c++

1) Overview of c++, OOPs concepts.
2) Differentiate gcc and g++ compiler.
3) Write a C++ program to read and display multiple values in a single I/O statement.
4) Write a program to perform division operation and print Quotient and Remainder based on user inputs.
5) Write a program to swap two numbers entered by the user without using a third variable.
6) Write a program to check if the year entered by the user is a leap year or not.
7) Write a program to print the maximum value out of three numbers.

Tools: Terminal (g++ Compiler).

EXPERIMENT-2

Aim: Implement the programs using concept of functions in c++.

1) Exchange values of two variables using call by value and call by reference (Pointer and reference variable both) concept.
2) Write UDF (User Defined Function) to convert lowercase characters of given string to uppercase and uppercase to lowercase.
3) Create inline functions for the following definition:
   a. Finding factorial of a given integer
   b. Finding cube of given integer
4) Using the concept of function overloading, find the perimeter of triangle, square and rectangle.
5) By overloading the print() function, display variable values of different data types.

Tools: Terminal (g++ Compiler).

EXPERIMENT-3

Aim: Implement the programs using concept of basic class and objects.

1) Create a class named student with data as roll no, name, cpi, etc.
    a. Create functions to display the relevant data of student class.
    b. Also write a function that can change/alter data of student class.
2) Define a class Employee with data employee name, city, basic salary, dearness allowance (DA) and house rent (HRA).
    a. Define getdata (), calculate (), and display () functions.
    b. Calculate() function should find the total salary and display() function should display it.
    Hint: Total = basic + basic * da / 100 + basic * hra / 100;
3) Create a class time with data as hours and minutes.
    a. Write a function gettime and puttime to read and display data of time class.
    b. Design a function called sum() that accepts two objects as arguments and displays the sum of two time objects.
4) Write a program that calculates sum, subtraction of two complex numbers using the concept of function returning object. Given functions should be declared as friendly functions.

Tools: Terminal (g++ Compiler).

EXPERIMENT-4

Aim: Implement the programs using concept of constructors and destructors

1) Write a Program to display the reverse of a number using the concept of constructor.
2) Using the concept of constructor overloading, find the area of Circle, rectangle, triangle.
3) Create a class distance with a data meter and centimeter. WAP to add two distances using the concept of copy constructor.
4) Write a program which demonstrates the use of constructor-destructors.

Tools: Terminal (g++ Compiler)

EXPERIMENT-5

Aim: Implement the programs using concept of operator overloading and string classes.

1) Write a program that calculates multiplication of two complex numbers by overloading multiplication sign.
2) Define distance class having data members feet and inches. Overload plus and minus operators for adding and subtracting two given distances respectively.
3) Perform following string operations using standard C++ string class functions:
   a. insert(),
   b. erase(),
   c. replace(),
   d. size(),
   e. compare(),
   f. length(),
   g. swap()

Tools: Terminal (g++ Compiler).

EXPERIMENT-6

Aim: Implement the programs using concept of inheritance.

1) Define one student class containing university and degree data. Define one employee class containing employee name and salary data. Now derive manager class from above class which contains all 4 details. Inheritance type should be private. All the classes have getdata() and showdata() functions to display the results.
2) Write a program which demonstrates the use of multi-level and multiple inheritance.

Tools: Terminal (g++ Compiler).

EXPERIMENT-7

Aim: Implement the programs using concept of polymorphism.

1) Write a program to demonstrate use of pointer with following concepts:
    - Arithmetic operations on pointers
    - Pointers with Arrays & Arrays of Pointers
    - Pointers to functions
    - Array of Pointers to Objects
2) Create a base class with two functions, display() and show(). Given function show() is a virtual function. Create a derived class which extends the base class.
3) Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape.
    - Add to the base class, a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements.
    - Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area.
      Area of rectangle = x * y
      Area of triangle = ½ * x * y

Tools: Terminal (g++ Compiler).

EXPERIMENT-8

Aim: Implement the programs using concept of I/O operations.

1) Write a program to read a list containing item name, item code, and cost interactively and produce a three column output as shown below:

NAME	CODE	COST
Turbo C++	1001	250.95
C++ Primer	905	95.70
……….	…..	……….
……….	…..	……….
……….	…..	……….

2) Write a program which reads a text from the keyboard and displays the following information on the screen in two columns:
   (a) Number of lines
   (b) Number of words
   (c) Number of characters

- Strings should be left-justified and numbers should be right-justified in a suitable field width.

Tools: Terminal (g++ Compiler).

EXPERIMENT-9

Aim: Implement the programs using concept of exception handling

1) Write a program that has divide() function which throws and catches division by zero exception.
2) Write a program that can throw integer, char and double exceptions in the same try block. Implement respective exception handling (multiple and single catch mechanism).
3) Write a program to demonstrate the use of ‘Rethrowing an Exception’ concept.

Tools: Terminal (g++ Compiler).

EXPERIMENT-10

Aim: Implement the programs using concept of templets.

1) Write a program to implement Bubble Sort using template functions.
2) Write a program which overloads template function with explicit function.
3) Write a class template to represent a generic vector. Include member functions to perform the following tasks:
   (a) To create the vector
   (b) To modify the value of a given element
   (c) To multiply by a scalar value
   (d) To display the vector in the following form (10, 20, 30 …)

Tools: Terminal (g++ Compiler).

LABWORK BEYOND CURRICULA

EXPERIMENT 11:

- Overview of File I/O concepts and program(s) using file stream classes.