

SECOND EDITION

# Lab Manual CS 161

## Introduction to Computer Programming C++

By Saqib Rasheed

2012-2013

Department of Computer Science  
Air University



Air University, PAF Complex, E-9, Islamabad

<b>Student Name</b>	
<b>Student Roll #</b>	
<b>Department</b>	
<b>Batch / Year</b>	

**NOTE:** At the end of this lab manual there is a Lab evaluation summary sheet. It is the responsibility of every student to get that sheet signed from his/her lab engineer after every lab. If He / She fail to do so, that particular unsigned lab will be marked zero.

To suggest more improvements and correction please feel free to write to [thesaqib@yahoo.com](mailto:thesaqib@yahoo.com)  
Copyright © 2015, 2007, All rights reserved. Manufactured in Pakistan. Except as permitted under the Pakistan States Copyright Act of 1971, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the writer. ([saqib.rasheed@mail.au.edu.pk](mailto:saqib.rasheed@mail.au.edu.pk))

## Contents

<b>INTRODUCTION.....</b>	<b>9</b>
<b>Lab Manual # 1 .....</b>	<b>11</b>
<b>Basic .....</b>	<b>11</b>
1.1 A typical C++ environment .....	12
1.2 Program No 1 .....	13
1.2.1 Out Put of Program No 1 .....	13
1.3 Declare Variables.....	13
1.4 Program No 2 Example Of Assignment Operator .....	13
1.4.1 Out Put of Program No 2 .....	14
1.5 Arithmetic Operators .....	14
1.5.1 Precedence .....	14
1.6 Program No 3.....	15
1.6.1 Code of Program No 3 .....	15
1.7 Program No 4.....	16
1.7.1 Code of Program No 4 .....	16
1.8 Program No 5.....	16
1.8.2 Code of Program No 5 .....	17
1.8.3 Out Put of Program No 5 .....	17
1.9 Program No 6.....	18
<b>Lab Manual # 2 .....</b>	<b>19</b>
<b>if / if-else Statements .....</b>	<b>19</b>
2.1 Que No 1 Largest value among three .....	20
2.1.1 Code Area .....	20
2.2 Que No 2 Grade Program using nested if else .....	21
2.3 Que No 3 vowel / Consonant .....	22
2.3.1 Code .....	22
2.4 Que No 4 Even / Odd.....	23
2.4.1 Code Area .....	23
2.5 Que No 5 Capital / Small Letter .....	24
2.5.1 Code Area .....	24
2.6 Que No 6 Ice / water / steam.....	25
2.6.1 Code Area .....	25

<b>Lab Manual # 3 .....</b>	<b>26</b>
<b>Switch Statement.....</b>	<b>26</b>
3.1 Que No 1 Switch statement.....	27
3.1.1 Code Area .....	27
<b>Lab Manual # 4 .....</b>	<b>30</b>
<b>For/while Loops.....</b>	<b>30</b>
4.1 Example .....	31
4.1.1 Output .....	31
4.2 Example Of While Loop.....	31
4.2.1 Output .....	31
4.3 Flow chart of while Loop.....	32
4.4 Factorial Definition .....	32
4.4.1 Code of Factorial Program.....	32
4.5 For loop.....	33
4.5.1 Example of for loop .....	33
4.5.2 Write the output of the program step by step.....	33
4.5.3 Final Output .....	33
4.6 Flow Chart of Table 2 .....	34
4.6.1 Write a program for the above flow chart.....	34
4.7 Example .....	35
4.8 QUE NO 1 Terminate program on entering zero .....	35
4.8.1 Code .....	35
4.9 QUE NO 2 Factorial program .....	36
4.9.1 Code .....	36
4.10 QUE NO 3 Fibonacci series.....	37
4.10.1 Code .....	37
4.11 Que No 4 . Armstrong number.....	38
4.11.1 Code .....	38
4.12 QUE NO 5 Largest among user defined numbers .....	39
4.12.1 Code .....	39
4.13 Nested loops Table 12 *12 .....	40
4.13.1 Code .....	40
4.14 Series using nested loops .....	41
4.14.1 Code for Series 1 .....	41

4.14.2 Code for series 2 .....	42
4.14.3 Code for series 3 .....	43
<b>Lab Manual # 5 .....</b>	<b>44</b>
<b>Do-while loop.....</b>	<b>44</b>
5.1 do-while loop syntax .....	45
5.1.1 Flow Chart of do-while loop.....	45
5.2 QUE 1 Calculate the sum of user defined numbers.....	46
5.2.1 Code .....	46
5.3 QUE NO 2 Calculate factorial of user defined numbers .....	47
5.3.1 Code .....	47
5.4 Que 3 Calculator .....	48
5.4.1 Code .....	48
<b>Lab Manual # 6 .....</b>	<b>49</b>
<b>Function .....</b>	<b>49</b>
6.1 Built In Function .....	50
6.1.1 Out Put .....	50
6.2 Write types of Function .....	50
6.3 Write Syntax Of function (Prototype, call & body of function) .....	50
6.4 Que No 1 square () function .....	51
6.4.1 Code .....	51
6.5 Que No 2 Finding average using function .....	52
6.5.1 Code .....	52
6.6 Que No 3 .....	53
6.7 Que No 4 finding area of rectangle.....	54
6.7.1 Code .....	54
6.8 Que No 5 Check date program.....	55
6.8.1 Code .....	55
6.9 Que No 6 Leap Year program.....	56
6.9.1 code .....	56
6.10 Que No 7 Finding largest using if else in function .....	57
6.10.1Code .....	57
6.11 QueNo 8.....	58
6.11.1 Code .....	58
6.12 Comparison of functions.....	59

6.13 Que No 9 Swap by using function .....	59
6.13.1 Code .....	59
6.14 Provide an Example of Functions Overloading .....	60
6.15 Que 10 Find Factorial from -1 to 10 using function .....	61
<b>Lab Manual # 7 .....</b>	<b>62</b>
<b>1-D Arrays / Strings .....</b>	<b>62</b>
7.1 Introduction .....	63
7.2 Que No 1 Displaying age of persons using array .....	63
7.2.1 Code .....	63
7.3 Que No 2 Changing values between arrays .....	64
7.3.1 Code .....	64
7.4 Que No 3 .....	65
7.4.1 Code .....	65
7.5 Que No 4 .....	66
7.5.1 Code .....	66
7.6 Que No 5 .....	67
7.6.1 Code .....	67
7.7 Que No 6 .....	68
7.7.1 Code .....	68
7.8 Que No 7 .....	69
7.8.1 Code .....	69
7.9 Que No 8 .....	70
7.10 Strings .....	71
7.10.1 Example of Strings .....	71
7.10.2 Que No 9 Get the name from user .....	72
7.11 The String I/O Function gets() & puts() .....	72
7.12 The String I/O Function gets() & puts() .....	72
7.13 strcpy & strcmp .....	73
7.13.1 strcpy & strcmp example .....	73
7.14 Que No 9 String Deletion .....	74
<b>Lab Manual # 8 .....</b>	<b>75</b>
<b>2-D Arrays .....</b>	<b>75</b>
8.1 Que No 1 Shows sale of Pharmaceutical distribution company .....	76
8.1.1 Code .....	76

8.2 Que no 2 Adding two matrices .....	77
8.2.1 Code .....	77
8.4 Que no 3 Printing matrix in reserve.....	78
8.4.1 Code .....	78
8.5 Que no 4 Transpose of a Matrix .....	79
8.5.1 Code .....	79
8.6 Que no 5 Agent Program .....	80
8.6.1 Code .....	80
<b>Lab Manual # 9 .....</b>	<b>81</b>
<b>Structures.....</b>	<b>81</b>
9.1 Introduction.....	82
9.2 Structure Example.....	82
9.2.1 Output .....	82
9.2.2 Description.....	83
9.3 Que No 1 Area of rooms.....	84
9.3.1 Code .....	84
9.4 Que No 2 Phone Number Program .....	85
9.4.1 Code .....	85
9.5 Que No 3 Employee record.....	86
9.5.1 Code .....	86
9.5 Que No 4 Memory size of a structure .....	87
9.5.1 Code .....	87
9.6 Que No 5 Average Age Program .....	88
9.6.1 Code .....	88
9.7 Que No 6 Nested Structure .....	89
9.7.1 Code .....	89
9.8 Que No 7 Access of structure data members with pointer to structure .....	90
<b>Lab Manual # 10 .....</b>	<b>91</b>
<b>Pointers .....</b>	<b>91</b>
10.1 Introduction to Pointers.....	92
10.1.1 Example of Pointers.....	92
10.2 Pointer To Arrays.....	92
10.2 Que No 1 Print the values from array .....	93
10.3 Que No 2 Print the values and memory address from an array .....	93

10.4 Pointer Arithmetic.....	94
10.4.1 Example of Pointer arithmetic .....	94
10.4.2 Output .....	94
10.5 Que No 3 Accessing values by Arithmetic operator.....	95
10.5.1 Code .....	95
10.6 Que No 4 Moving in array through pointers.....	96
10.6.1 Code .....	96
10.7 Pointer Comparison .....	96
10.7.1 Pointer Comparison Example .....	97
10.8 Que No 4 Question Max .....	97
10.8.1 Code .....	97
10.9 Pointer to functions .....	98
10.9.1 Que No 5 Swap the same values using pointers. ....	98
10.10 Que No 6 Returning more than one values from a function .....	99
10.10.1 Code .....	99
<b>Lab Manual # 11 .....</b>	<b>100</b>
<b>Files.....</b>	<b>100</b>
11.1 Introduction to Files .....	101
11.2 Files of C.....	101
11.3 Files in C++ .....	102
11.3.1 Open a file.....	102
11.4 Que No 1 Create a text file .....	103
11.4.1 Code .....	103
11.5 Que No 2 Read from the file.....	104
11.5.1 Code .....	104
11.6 Que No 3 Write data through variable .....	105
11.6.1 code .....	105
11.7 String with Embedded blanks .....	106
11.7.1 Code .....	106
<b>Lab Evaluation Summary .....</b>	<b>107</b>



## INTRODUCTION

The objective of this lab manual is to give students step-by-step examples to become familiar with programming concepts, design, and coding.

## FEATURES

To ensure a successful experience for instructors and students alike, these lab manuals include the following features:

- **Lab Objectives**—Every lab has a brief description and list of learning objectives
- **Materials Required**—Every lab includes information on hardware, software, and other materials you will need to complete the lab
- **Completion Times**—Every lab has an estimated completion time so that you can plan your activities more accurately
- **Activity Sections**—Labs are presented in manageable sections; where appropriate, additional Activity Background information is provided to illustrate the importance of a particular project
- **Step-by-Step Instructions**—Every lab provides steps to enhance technical proficiency; some labs include Critical Thinking exercises to challenge students
- **Review Questions**—Some labs include review questions to help reinforce concepts presented in the lab

## SOFTWARE REQUIREMENTS

- Computer running Windows 98, Windows Me, Windows 2000, Windows XP
- Recommended compiler is Microsoft Visual C++ .NET or Microsoft Visual Studio

## COMPLETING THE LAB ASSIGNMENTS

Some lab assignments require written answers to complete an exercise, while others are programming assignments that require you to work with a C++ compiler.

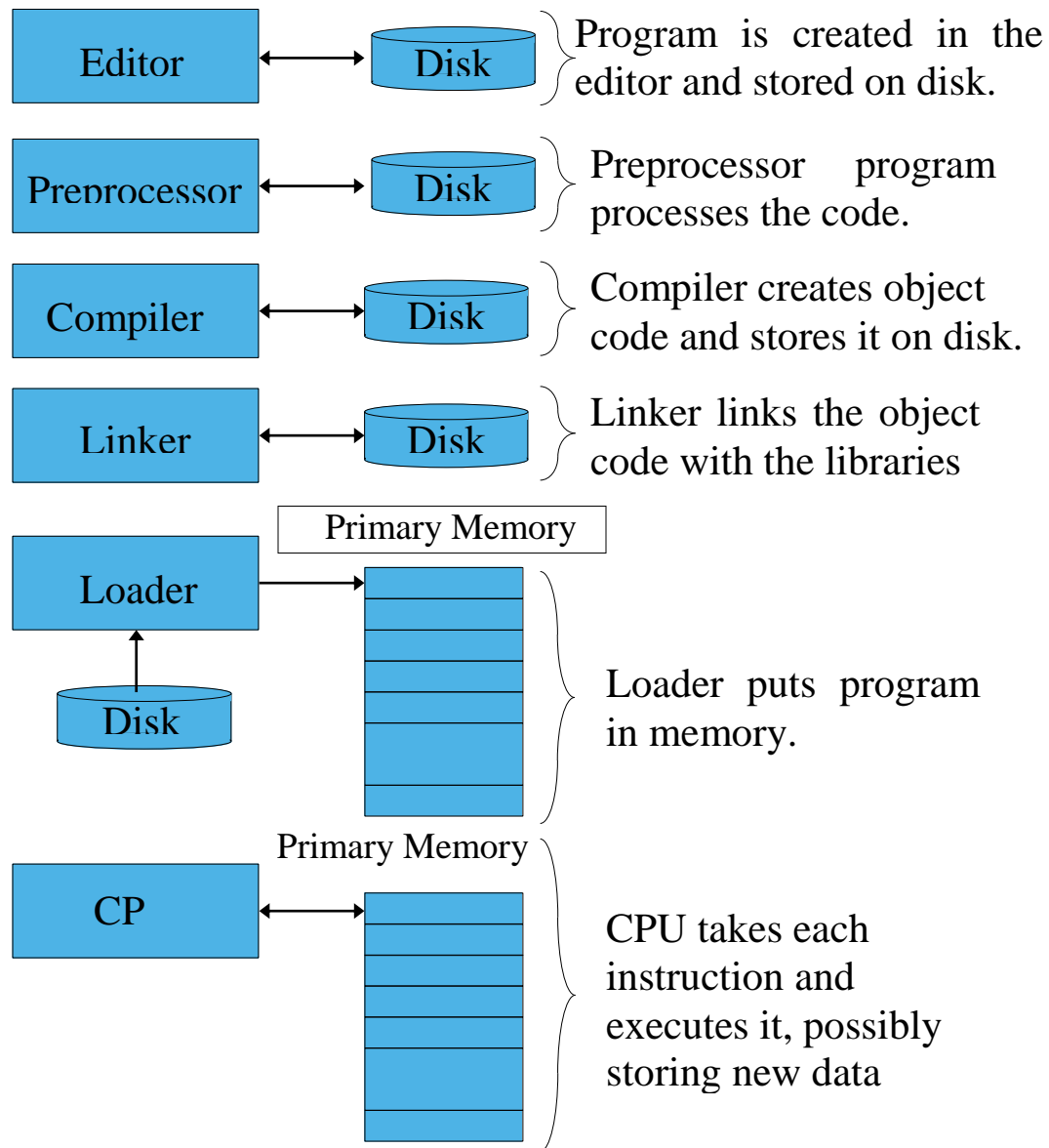
- Check with your instructor for instructions on completing the written assignments. For example, you can print pages directly from the appropriate PDF file, and then write directly on the page.
- To complete the programming assignments, use the compiler that your instructor recommends or requires. Print all the documentation assigned, including program code, program prompts, input, and output displayed on the screen, input files, and output files. You can submit your written answers and the printed documentation with a lab cover sheet for grading. If your instructor requires an electronic copy of your work, e-mail the completed assignment to your instructor or include a removable disk with your work. Your instructor will tell you what is

needed, but be sure to submit the .cpp, .h, and any .srt or .txt files that you create, as well as any input and output files. Also include your name or ID in the titles of all your files. To provide program documentation, compile and run your program, copy the prompts, input, and output (if appropriate), and paste them as a block comment at the end of your program. Use the Copy and Paste features of your C++ program development kit to do so. After you paste the comment in the program, either print the program file from your text editor or submit the program file to your instructor electronically.

# **Lab Manual # 1**

# **Basic**

## 1.1 A typical C++ environment



**1.2 Program No 1**

```
#include <iostream.h>
main ( )
{
    cout << " Welcome to Air University ";
}
```

**1.2.1 Out Put of Program No 1****1.3 Declare Variables**

Declare 3 Integer Type & 3 float type Variables.

INT	FLOAT

**1.4 Program No 2 Example Of Assignment Operator**

```
#include <iostream>
using namespace std;
int main ()
{
    int a, b;
    a = 10;
    b = 4;
    a = b;
    b = 7;
    cout << "a :";
    cout << a;
    cout << " b :";
    cout << b;
    return 0;
}
```

**1.4.1 Out Put of Program No 2**

--

**1.5 Arithmetic Operators**

Write Arithmetic Operators


**1.5.1 Precedence**

Write precedence of the arithmetic operators




### 1.7 Program No 4

Write a program in C++ which prints following output,

```
$*****$
$*****$
$*   Welcome to Air University   *$
$*   School of Engineering       *$
$*           Islamabad.         *$
$*****$
$*****$
```

#### 1.7.1 Code of Program No 4

### 1.8 Program No 5

Write a program in C++, which takes radius from the user and calculate the area of sphere i.e

Area=4pr<sup>2</sup>

(Hint p = 3.1416

Area = 4 \* 3.1416 \* r \* r)



### 1.8.2 Code of Program No 5

### 1.8.3 Out Put of Program No 5

**1.9 Program No 6**

Write a program to find the number of bytes occupied by various data types using the sizeof operator?

```
int a;  
char b;  
float c;  
long int d;  
bool e;  
unsigned int j;  
unsigned long k;
```

**1.9.1 Code**

# **Lab Manual # 2**

## **if / if-else Statements**

## 2.1 Que No 1 Largest value among three

Write a program in C++ that take input of three integer's numbers from user. Find the largest number among three of them.

### 2.1.1 Code Area

## 2.2 Que No 2 Grade Program using nested if else

Write a program in C++ using if/else operator with nested statements to find the grade of a student.

The detail is as follow.

grade >= 90 → Grade A  
grade >= 80 → Grade B  
grade >=70 → Grade C  
grade >=60 → Grade D

### 2.2.1 Code Area

### 2.3 Que No 3 vowel / Consonant

Write a program in C++ to input a single character and print a message "It is vowel" if it is vowel otherwise print message "It is a consonant" Use if-else structure and OR (||) operator only

#### 2.3.1 Code

**2.4 Que No 4 Even / Odd**

Write a Program in C++ that take an Integer Value from the user and tell that the number is EVEN or ODD.

**2.4.1 Code Area**

**2.5 Que No 5 Capital / Small Letter**

Write a program in C++ that take a single character from the user, and tells it's a Small Letter or it's a CAPITAL letter using nested if statement only.

**2.5.1 Code Area**



**2.6 Que No 6 Ice / water / steam**

Make a program in C ++ that tells the form of Water whether it is Ice, Water or Steam. Display the menu also as under.

Temperature Less than 0 = ICE

Temperature Greater than 0 & Less than 100 = Water

Temperature Greater than 100 = STEAM

**2.6.1 Code Area**

# Lab Manual # 3

# Switch Statement

### 3.1 Que No 1 Switch statement

Write a program in C++ using switch statement that contain option as under

Enter 1--> To Find Largest Number Among Three Variables.

Enter 2--> To Find ODD or EVEN

Enter 3--> To Find Condition of Water

Enter 4--> To Find Grade Of Student

#### **Detail of Option 3**

Temperature Less than 0 = ICE

Temperature Greater than 0 & Less than 100 = Water

Temperature Greater than 100 = STEAM

#### **Detail of option 4**

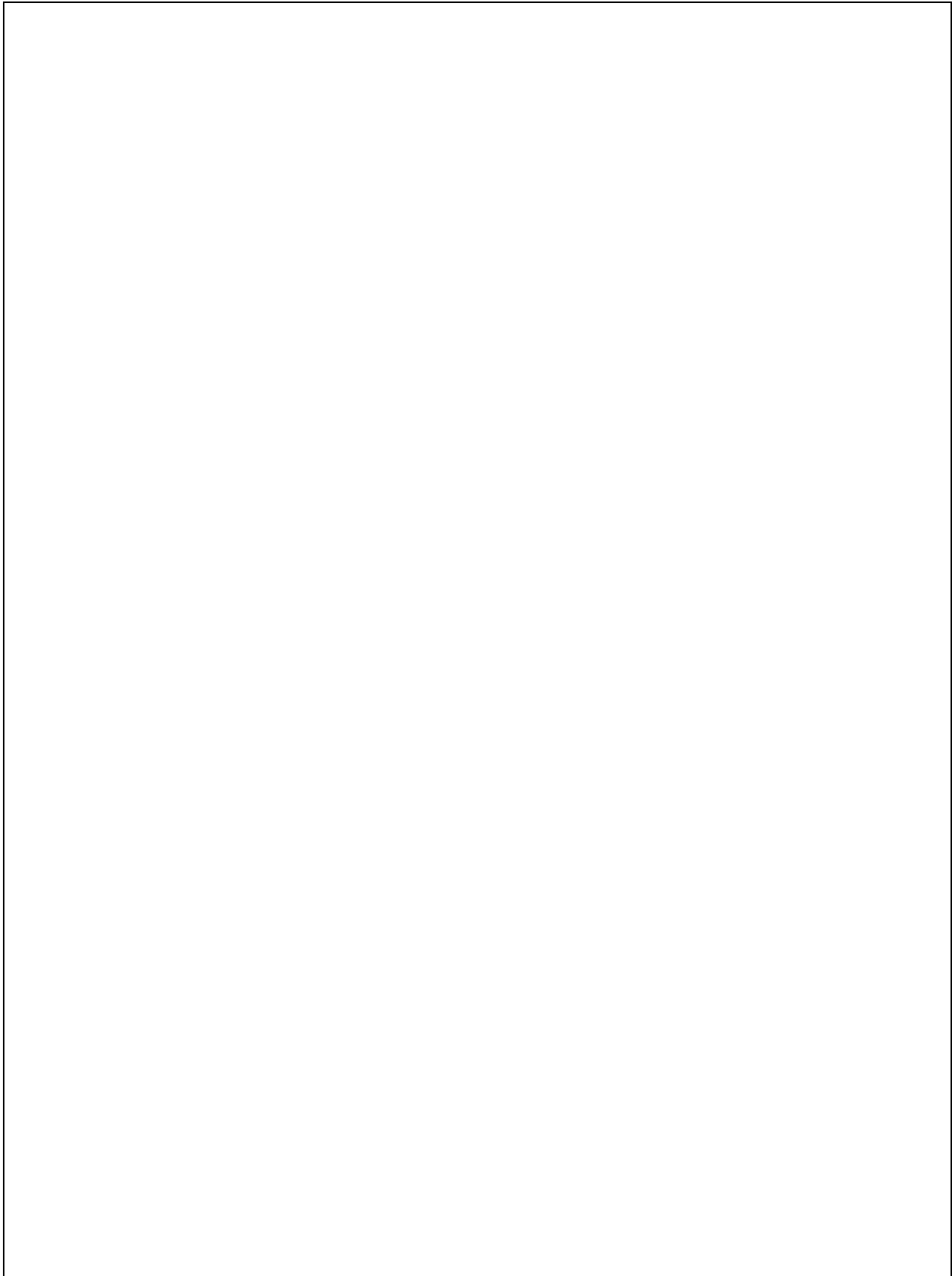
grade >= 90 → Grade A

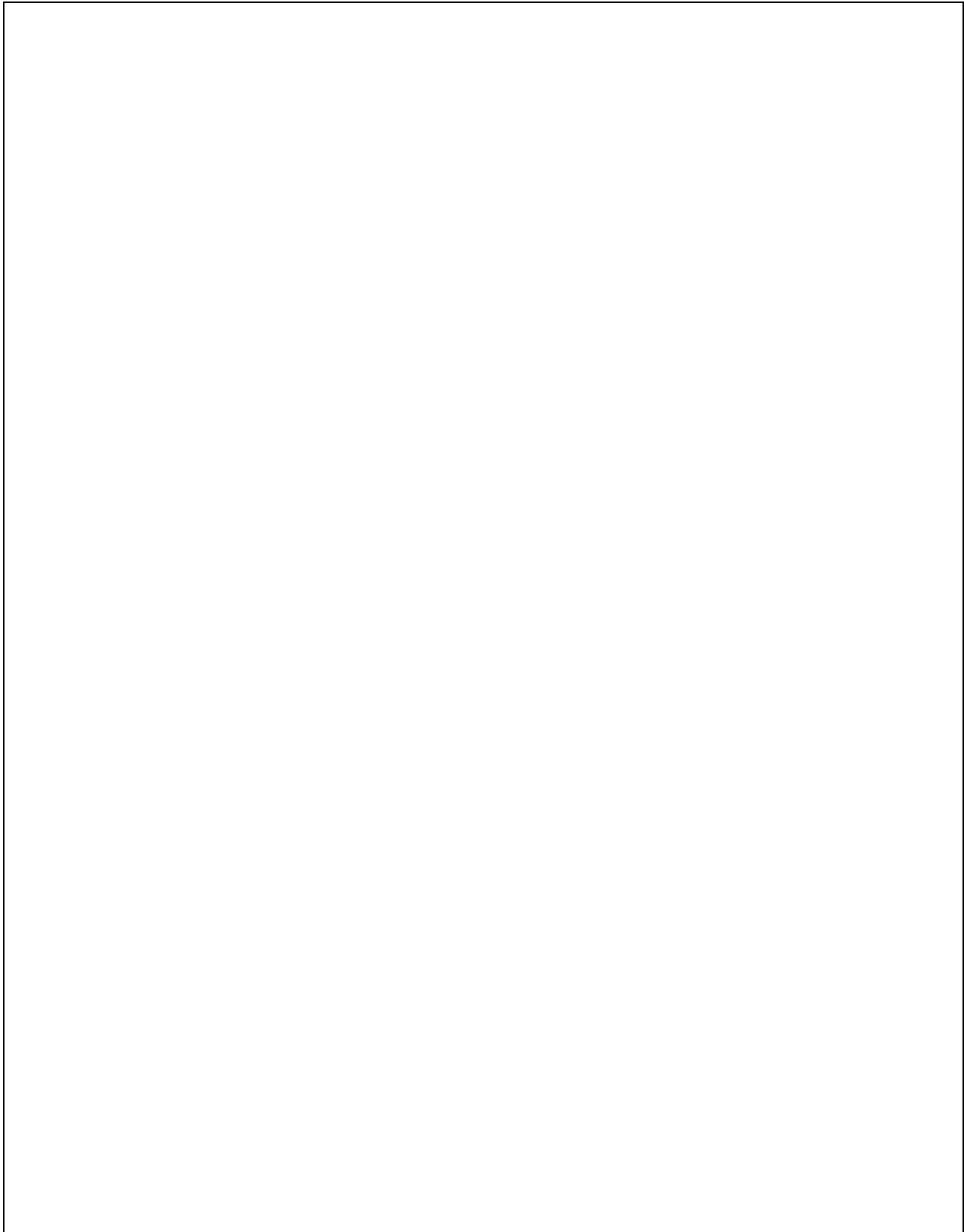
grade >= 80 → Grade B

grade >=70 → Grade C

grade >=60 → Grade D

#### 3.1.1 Code Area





# Lab Manual # 4

## For/while Loops

## 4.1 Example

```
int sum ;  
sum = 1+2+3+4+5+.....+10 ;  
cout << sum ;
```

### 4.1.1 Output

## 4.2 Example Of While Loop

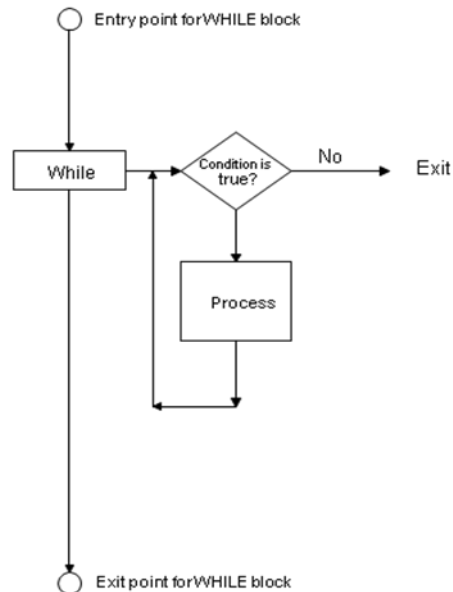
```
int sum , number ;  
sum = 0 ;  
number = 1 ;  
while ( number <= 1000 )  
{  
    sum = sum + number ;  
    number = number + 1 ;  
}  
cout << " The sum of the first 1000 integer starting from 1 is " << sum ;
```

### 4.2.1 Output

### 4.3 Flow chart of while Loop

## Flow Chart for While Construct

### WHILE Statement



### 4.4 Factorial Definition

$$n! = n * (n-1) * (n-2) * (n-3) \dots \dots \dots * 3 * 2 * 1$$

#### 4.4.1 Code of Factorial Program

```

int number ;
int factorial ;
factorial = 1 ;

cout << "Enter the number of Factorial" ;
cin >> number ;
while ( number >= 1 )
{
    factorial = factorial * number ;
    number = number - 1 ;
}
cout << "Factorial is" << factorial ;
  
```

### Out Put




## 4.5 For loop

```
for ( initialization condition ; termination condition ; increment condition )
{
    statement ( s ) ;
}
```

### 4.5.1 Example of for loop

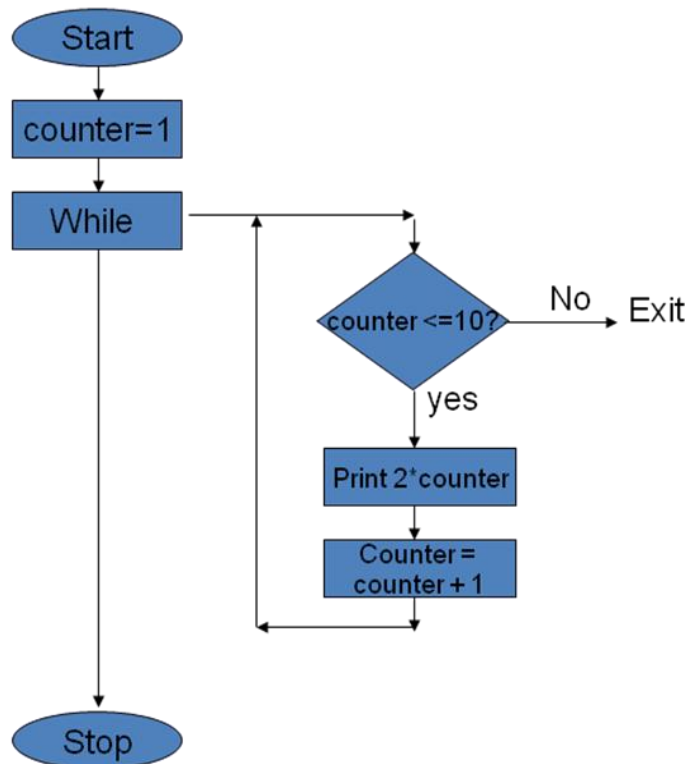
```
int counter ;
for( counter = 0 ; counter < 10 ; counter = counter + 1 )
    cout << counter;
```

#### 4.5.2 Write the output of the program step by step

[illegible]

### 4.5.3 Final Output

--

**4.6 Flow Chart of Table 2****Flow chart for the 'Table' example****4.6.1 Write a program for the above flow chart**

**4.7 Example**

Write a Program That Display numbers from  
1,2,3,4,5,6,7,8,9.....10000

**4.7.1 Code****4.8 QUE NO 1 Terminate program on entering zero**

Write a Program in C++ that Exit on entering 0, Using While Loop.

**4.8.1 Code**

**4.9 QUE NO 2 Factorial program**

Write a Program in C++ that calculate the factorial of a user defined number, using for loop

**4.9.1 Code**

**4.10 QUE NO 3 Fibonacci series**

Write a Program in C++ that shows The Fibonacci series

1 1 2 3 5 8 13 ..... Using while loop

**4.10.1 Code**

**4.11 Que No 4 . Armstrong number**

Write a Program in C++ that check the user defined number is Armstrong or not.i.e 153 is an Armstrong number

$$\begin{array}{rccccccc} (1)^3 & + & (5)^3 & + & (3)^3 & & \\ 1 & + & 125 & + & 27 & = & 153 \end{array}$$

**4.11.1 Code**

**4.12 QUE NO 5 Largest among user defined numbers**

Write a Program in C++ that takes Integer values from user, and then find the largest number among all the Integers, and display the largest number on the screen. Use While loop to control the Input, Terminate the Program on entering zero. Find the Largest Integer Using If Statement.

**4.12.1 Code**

**4.13 Nested loops Table 12 \*12**

Write a program in C++ that prints a tables  
Starting from 1 →12. i.e

1	2	3	4	5	6	7	8	9	10	11	12
2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	27	30	33	36
4	8	12	16	20	24	28	32	36	40	44	48
5	10	15	20	25	30	35	40	45	50	55	60
6	12	18	24	30	36	42	48	54	60	66	72
7	14	21	28	35	42	49	56	63	70	77	84
8	16	24	32	40	48	56	64	72	80	88	96
9	18	27	36	45	54	63	72	81	90	99	108
10	20	30	40	50	60	70	80	90	100	110	120
11	22	33	44	55	66	77	88	99	110	121	132
12	24	36	48	60	72	84	96	108	120	132	144

**4.13.1 Code**



**4.14 Series using nested loops**

Develop a code in C++ that generate the following series .Use nested while loop!

**Series No.1**

1  
1 2  
1 2 3  
1 2 3 4

**Series No.2**

1  
2 2  
3 3 3  
4 4 4 4

**Series No.3**

1  
2 3  
4 5 6  
7 8 9 10

**4.14.1 Code for Series 1**

#### 4.14.2 Code for series 2

#### 4.14.3 Code for series 3

# Lab Manual # 5

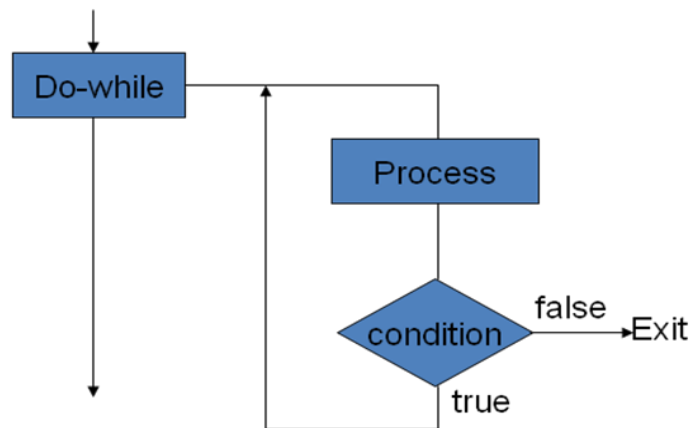
## Do-while loop

## 5.1 do-while loop syntax

```
do  
{  
    statements ;  
}  
while ( condition ) ;
```

### 5.1.1 Flow Chart of do-while loop

## Flow chart for do-while loop



## 5.2 QUE 1 Calculate the sum of user defined numbers

Develop a program in C++ that take integer type values from user & calculate the sum from '0' to number entered, using do while loop.

*Number = 5*

$$1+2+3+4+5 = 15$$

### 5.2.1 Code

**5.3 QUE NO 2 Calculate factorial of user defined numbers**

Develop a program in C++ that calculate the factorial of a given number. Use for loop to calculate the factorial, & do – while loop to perform the operation as many times as user want.

**5.3.1 Code**

### 5.4 Que 3 Calculator

Create the equivalent of a four-function calculator. The program should Ask the user to enter a number, an operator, and another number (10 + 20), using floating point. It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. Use switch statement to select the operation. Finally, display the Result.

#### 5.4.1 Code



# Lab Manual # 6

# Function

## 6.1 Built In Function

```
#include <cmath> // defines the sqrt() function
#include <iostream.h>
```

```
int main()
{ // tests the sqrt() function:
    for (int x=0; x <=9; x++)
        cout << "\t" << x << "\t" << sqrt(x) << endl;
}
```

### 6.1.1 Out Put

--

## 6.2 Write types of Function


## 6.3 Write Syntax Of function (Prototype, call & body of function)

--

**6.4 Que No 1 square () function**

Write a program in C++ that has a function of square (). The user should enter the number and the code should display the square of the given number.

**6.4.1 Code**

**6.5 Que No 2 Finding average using function**

Get three integers numbers from user, Pass them to function, Add them and find the average in function body, Display the Average in function body

**6.5.1 Code**

**6.6 Que No 3**

Write a code that take numbers from user and displays its cube. The Code should reads integers and prints their cubes until the user inputs the sentinel value 0. Each integer read should be passed to the cube() function by the call cube(n). The value returned by the function should replaces the expression cube(n) and then should be passed to the output object cout.

**6.6.1 Code**

**6.7 Que No 4 finding area of rectangle**

Write a function that finds the area of the rectangle on providing length and width.  
Get Length & width from user in main() Call the function area() Calculate the length and return the area Display the result in main()

**6.7.1 Code**

**6.8 Que No 5 Check date program**

Get month ,day & year from user in main() Call the function printDate(int, int, int), Put a check in printDate() using if statement (month < 1 || month > 12 || day < 1 || day > 31 || year < 0) & if it violates the rule display "*Must Enter a Valid Date*", Using switch Statement get the month, Day and year are displayed normally, Termination should be on entering 0 in months

**6.8.1 Code**

### 6.9 Que No 6 Leap Year program

A leap year is a year in which one extra day (February 29) is added to the regular calendar. Most of us know that the leap years are the years that are divisible by 4. For example, 1992 and 1996 are leap years. Most people, however, do not know that there is an exception to this rule: centennial years are not leap years. For example, 1800 and 1900 are not leap years. Furthermore, there is an exception to the exception: centennial years which are divisible by 400 are leap years. Thus, the year 2000 is a leap year.

- Make a program that full fills the above criteria using functions
- The program should terminate on entering 0
- The return type should be Boolean i.e `bool isLeapYear(int);`

#### 6.9.1 code



**6.10 Que No 7 Finding largest using if else in function**

Write a program in C++ that take two numbers from user and find the largest among two using function.

➤ The Conditions are

*if (a==b)*

*"A, B are the same"    a = b, (values)*

*else if (a < b)*

*"A & B are not same "    a != b (values)*

*" A is less than B "    a < b (values)*

*else*

*"A is Greater than B "    a > b (values)*

**6.10.1Code**

**6.11 QueNo 8**

Develop a program in C++ that has function printTempOpinion() which prints "Cold" on if the temperature is below 10, "OK" if the temperature is in the range 20 -> 30, "Hot" if the temperature is above 30.

**6.11.1 Code**



**6.14 Provide an Example of Functions Overloading**

**6.15 Que 10 Find Factorial from -1 to 10 using function**

# **Lab Manual # 7**

## **1-D Arrays / Strings**

## 7.1 Introduction

They are special kind of data type In C++ each array has

- ✓ name
- ✓ data type
- ✓ size
- ❖ They occupy continuous area of memory

## 7.2 Que No 1 Displaying age of persons using array

Write a program in C++ that take age of five persons and then just display the age of each person by using arrays.

### 7.2.1 Code

**7.3 Que No 2 Changing values between arrays**

Develop a Program that takes array elements from user and then transfer those elements to another array. Size of array will be 10

**7.3.1Code**



**7.4 Que No 3**

Develop a program that takes 5 array elements from user. Swap position [2] element with position [4] element.

**7.4.1 Code**

**7.5 Que No 4**

Write a program to input data into two different arrays and then to add the two arrays and store the result in the third array.

**7.5.1 Code**

**7.6 Que No 5**

Write a program in C++, to input data into an array. Enter a value from the Keyboard and find out the location of the entered value in the array. If the entered number is found in the array, display the message "Number Found" else display "Number Not Found"

**7.6.1 Code**

**7.7 Que No 6**

Write a program that takes input from user and checks if the word is Palindrome or Not

**7.7.1Code**

**7.8 Que No7**

Find the Average of 10 numbers passed to an array, using function. The Array should be controlled by while loop.

**7.8.1 Code**

**7. 9 Que No 8**

Write a Program in C++ that do the bubble sorting in Array.

Hint

- ✓ Array Elements are entered by user in “main function”, the elements are user Defined. Terminate on “zero”
- ✓ Make a separate function “*sort()*” to do the bubble sort.
- ✓ Pass array elements to function “*sort()*”.
- ✓ Function should sort the array elements.
- ✓ Display the sorted array elements in main.

**7.9.1 Code**

## 7.10 Strings

A string constant is a one-dimensional array of characters terminated by a null ( '\0' ). For example,

```
char name[ ] = { 'H', 'A', 'E', 'S', 'L', 'E', 'R', '\0' } ;
```

### 7.10.1 Example of Strings

```
/* Program to demonstrate printing of a string */
main( )
{
char name[ ] = "Klinsman" ;
int i = 0 ;
while ( i <= 7 )
{
    cout<<name[i];
    i++ ;
}
}
```

```
/* Program to demonstrate printing of a string */
main( )
{
char name[ ] = "Klinsman" ;
int i = 0 ;
while (name[i] !='\0')
{
    cout<<name[i];
    i++ ;
}
}
```

**7.10.2 Que No 9 Get the name from user**

Write a program that gets the name from user and then print back on the screen.

**7.11 The String I/O Function gets() & puts()**

cin & cout are not so versatile because they ignore characters after white space, so to overcome this problem C/C++ uses puts & gets function from library file <stdio.h>

**7.12 The String I/O Function gets() & puts()**

Write a program that gets the name from user and then print back on the screen using gets and puts functions.



### **7.13 strcpy & strcmp**

#### **7.13.1 strcpy & strcmp example**

**7.14 Que No 9 String Deletion**

Write a program in C++ that take string from user and then enter the position to delete.  
Make a separate function for string deletion

# Lab Manual # 8

## 2-D Arrays

**8.1 Que No 1 Shows sale of Pharmaceutical distribution company**

A Distributor of a Pharmaceutical Company has 4 Districts, for supply the medicine. He requires a program that can display the sales of all his Districts. Write a Program in C++ Using two Dimensional Array that shows the Following Output. The Program should display the Sale, Districts wise and up to Months i.e

**8.1.1Code**

**8.2 Que no 2 Adding two matrices**

Write a program in C++ that take two matrices and then Add them After inserting two matrices first display the both matrices and then add them and show the result. i.e

**8.2.1 Code**

**8.4 Que no 3 Printing matrix in reserve.**

Enter the values in a matrix and print it in reverse Column order

**8.4.1 Code**

**8.5 Que no 4 Transpose of a Matrix**

Write a Program in C++ that Display the Transpose of a Matrix.

**8.5.1Code**

**8.6 Que no 5 Agent Program**

Write a program in C++, which take Agent code (123,258,..) and Traveling expense (Rs = 5000, 6000,...) of the agent. Find the agent who had spent most money in all, Display the agent code and amount after searching in 2 D Array.

**8.6.1 Code**



# Lab Manual # 9

# Structures

## 9.1 Introduction

- A Structure is a collection of simple variables. The Variables in a structure can be of different types. Some can be int, some can be float, and so on.
- The data items in a structure are called the members of the structure.
- The structure is a kind of blue print specifying what information is necessary for a single part.

## 9.2 Structure Example

```
struct part
{
    int modelnumber;
    int partnumber;
    float cost;
};
int main ()
{
    part part1, part2;
    part1.modelnumber = 1111;
    part1.partnumber = 111;
    part1.cost = 111.11;

    part2.modelnumber = 222;
    part2.partnumber = 2222;
    part2.cost = 222.222;
    cout<<"\nModel of Part1 = "<<part1.modelnumber;
    cout<<"\nPart of part 1 = "<<part1.partnumber;
    cout<<"\nCost of part1 = "<<part1.cost<<endl;

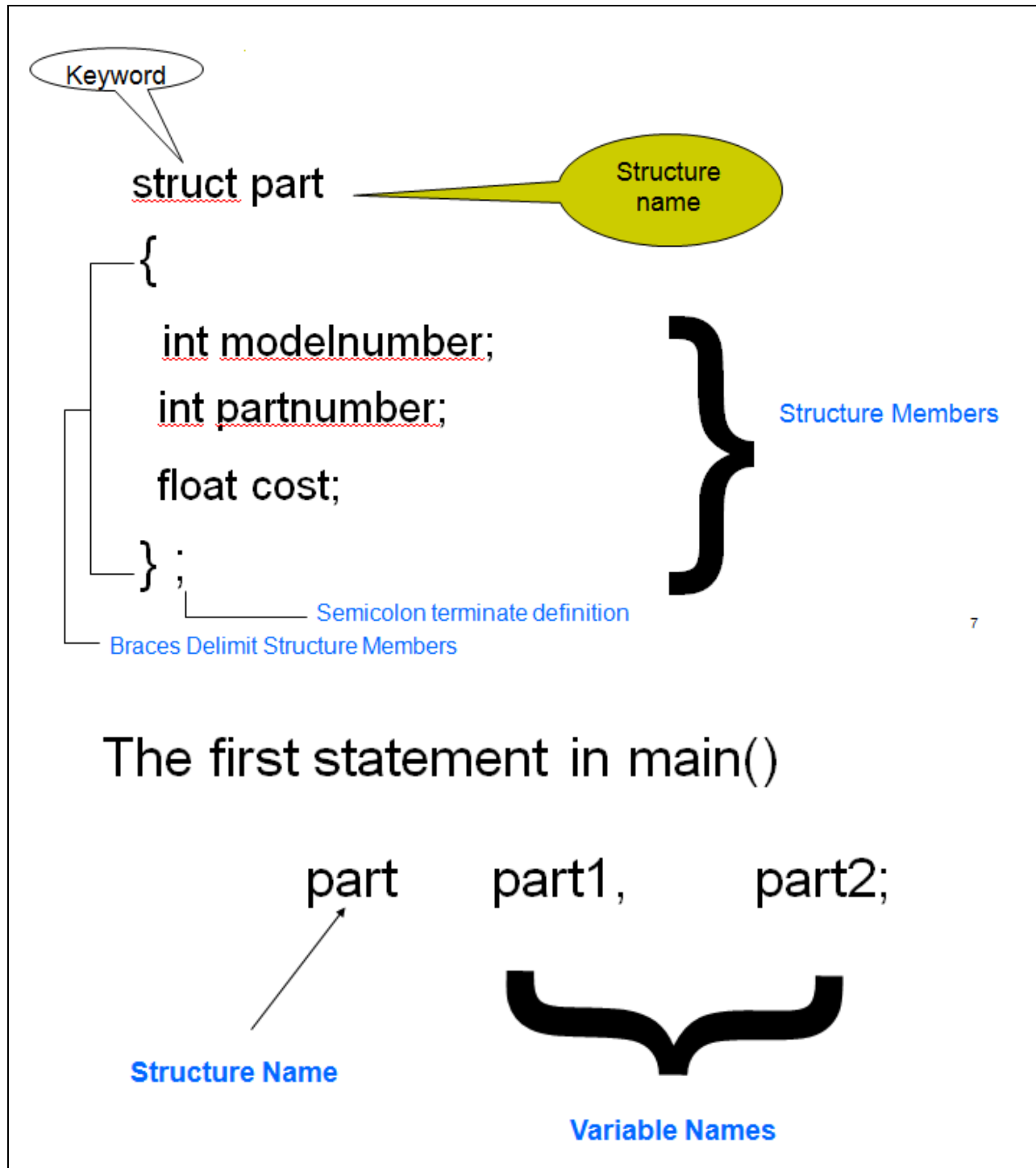
    cout<<"\nModel of part2 = "<<part2.modelnumber;
    cout<<"\nPart of part2 = "<<part2.partnumber;
    cout<<"\nCost of part2 = "<<part2.cost<<endl;
    return 0;
}
```

### 9.2.1 Output

```
Model of Part1 = 1111
Part of part 1 = 111
Cost of part1 = 111.11
Model of part2 = 222
Part of part2 = 2222
Cost of part2 = 222.222
```

### 9.2.2 Description

The structure definition serves only as a blueprint for the creation of variables of type part. It does not itself create any structure variables; that is, it does not set aside any space in memory or even name any variables. This is unlike the definition of a simple variable, which does set aside memory.



**9.3 Que No 1 Area of rooms**

Write a program in C++ that shows the area of 3 room's. Using Structure namely "distance". Take input of feet & inches from user for variable d1 (feet & inches), assign variable d2 = {10, 5.25} values. Now add feet and inches of d1 & d2 and store in d3.

Display d1 (feet & inches) d2 (feet & inches) d3 (feet & inches) separately. Put Condition if d1 & d2 inches increase by 12 it become a foot.

**9.3.1 Code**

### 9.4 Que No 2 Phone Number Program

A phone number, such as 212- 767-8900, can be thought of as having three parts: the area code (212), the exchange (767), and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structures variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers. The interchange might look like this:

#### 9.4.1 Code

**9.5 Que No 3 Employee record**

Create a structure called emp that contains three members,  
*int id, char name[100], float sal.*

Ask the user to fill in data for three employees and then display information for each employee.

**Hint**

- Variable of struct emp will be array
- Use while / for loop to control array

**9.5.1 Code**

**9.5 Que No 4 Memory size of a structure**

Write a program using “sizeof()” function that calculate the size of structure

**9.5.1 Code**

**9.6 Que No 5 Average Age Program**

Write a program to calculate the average age and average GPA of a class having 10 students. Also determine the grade of the class and the student with max GPA. Use a student structure and manipulate it to get the desired result.

**9.6.1 Code**



**9.7 Que No 6 Nested Structure**

Write a program that contains nested structure

**9.7.1 Code**

### 9.8 Que No 7 Access of structure data members with pointer to structure

# **Lab Manual # 10**

# **Pointers**

## 10.1 Introduction to Pointers

```
int *myptr ;  
    myptr is pointer to an integer
```

### 10.1.1 Example of Pointers

```
#include<iostream.h>  
void main ()  
{  
    int var1 = 10;  
    int var2 = 20;  
    int var3 = 30;  
    cout<<&var1<<endl  
        <<&var2<<endl;  
    int *ptr;  
    ptr = &var1;  
    cout<<ptr<<endl;  
    ptr = &var2;  
    cout<<ptr<<endl;  
}
```

## 10.2 Pointer To Arrays

```
#include<iostream.h>  
void main ()  
{  
    int array[5] = {31,54,77,52,93};  
    for(int j =0; j<5; j++)  
    {  
        cout<<array[j]<<endl;  
    }  
}
```

```
#include<iostream.h>  
void main ()  
{  
    int array[5] = {31,54,77,52,93};  
    int* ptr;  
    ptr = array;  
    for(int j =0; j<5; j++)  
    {  
        cout<<*(ptr++)<<endl;  
    }  
}
```

**10.2 Que No 1 Print the values from array**

Write a program that prints the values from an array using pointer variable. The array is given below

```
int y [ 10 ]= {6,2,3,12};
```

**10.3 Que No 2 Print the values and memory address from an array**

Write a program that prints the values from an array using pointer variable. The array is given below

```
int y [ 10 ]= {6,2,3,12};
```

## 10.4 Pointer Arithmetic

```
int x = 10 ;  
int *yptr ;  
yptr = &x ;  
*yptr += 3 ;  
yptr += 3 ;
```

### 10.4.1 Example of Pointer arithmetic

```
long* pnumber = NULL;  
long number1 = 10, number2 = 20;  
  
pnumber = &number1;  
*pnumber += 2;  
  
cout<<"\nnumber1 = "<<number1  
    <<" &number = "<<pnumber;  
pnumber = &number2;  
number1 = *pnumber *4;  
  
cout<<"\nnumber1 = "<<number1  
    <<" pnumber = "<<pnumber  
    <<"pnumber = "<<*pnumber;
```

### 10.4.2 Output

```
number1 = 12 &number = 0x0012FF78  
number1 = 80 pnumber = 0x0012FF74  
pnumber = 20  
Press any key to continue
```

**10.5 Que No 3 Accessing values by Arithmetic operator**

Write a program that displays the values using pointer variable from an array given below using Arithmetic Increment operator .

```
int y[5]={22,33,44,55,66};
```

**10.5.1 Code**

**10.6 Que No 4 Moving in array through pointers**

Write a program that display only 6<sup>th</sup> element of an array given below using pointers.

```
int y [10] ={11,22,33, 44,55,66,77,88,99,110}
```

**10.6.1 Code****10.7 Pointer Comparison**

```
if ( y1 > y2 )  
if ( y1 >= y2 )  
if ( y1 == y2 )
```

```
if ( *y1 > *y2 )
```



**10.7.1 Pointer Comparison Example**

```
int y [10]={11,22,33,44,55,66,77,88,99,110} ;

int *y1, *y2;
y1= &y[0];
y2= &y[3];;

cout <<"\n Y1= "<<*y1;
cout <<"\n Y2= "<<*y2;

if (*y1 < *y2)
    cout<<"\nY1 is Smaller"<<endl;
else
    cout<<"\nY2 is smaller"<<endl;
```

**10.8 Que No 4 Question Max**

Write a program that take two numbers an input from user. Find the maximum from both of them using the dereference pointer comparison

**10.8.1 Code**

**10.9 Pointer to functions**

```
main( )
{
int a = 10, b = 20 ;
swapv ( a, b ) ;
cout<<"\na = "<<a<<" b= "<<b;
}
swapv ( int x, int y )
{
int t ;
t = x ;
x = y ;
y = t ;
cout<<"\nx = "<<x<<" y = "<<y;
}
```

The above given code is swapping the values without pointers.

**10.9.1 Que No 5 Swap the same values using pointers.**

**10.10 Que No 6 Returning more than one values from a function**

Write a program that gets the radius from user, pass radius to a function areaperi() and function areaperi() returns "area" and "perimeter" by reference

Using a call by reference intelligently we can make a function return more than one value at a time, which is not possible ordinarily.

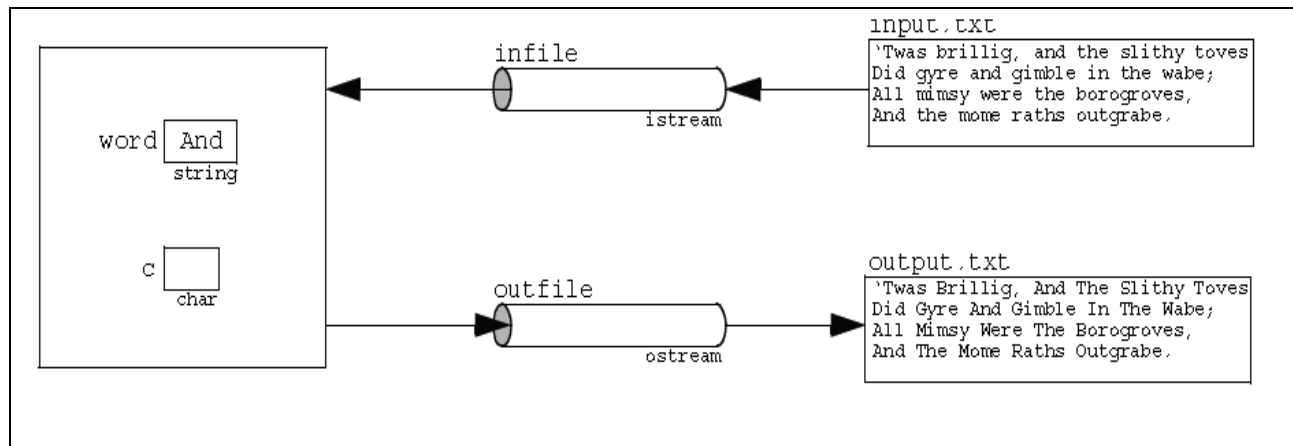
**10.10.1 Code**

# **Lab Manual # 11**

## **Files**

## 11.1 Introduction to Files

File processing in C++ is very similar to ordinary interactive input and output because the same kinds of stream objects are used. Input from a file is managed by an ifstream object the same way that input from the keyboard is managed by the istream object cin. Similarly, output to a file is managed by an ofstream object the same way that output to the monitor or printer is managed by the ostream object cout. The only difference is that ifstream and ofstream objects have to be declared explicitly and initialized with the external name of the file which they manage. You also have to #include the <fstream> header file (or <fstream.h> in pre-Standard C++) that defines these classes



Notice that the program has four objects: an ifstream object named infile, an ofstream object named outfile, a string object named word, and a char object named c. The advantage of using external files instead of command line redirection is that there is no limit to the number of different files that you can use in the same program.

## 11.2 Files of C

```
//Program that create / open a file
#include<stdio.h>
#include<conio.h>
void main ()
{
    FILE *fptr;
    char ch;
    fptr = fopen("first.txt","w");
    while((ch =getche()) != '\r')
        putc(ch,fptr);
    fclose(fptr);
}
```

### 11.3 Files in C++

```
include <fstream>
using namespace std;
int main ()
{
    ofstream myfile;
    myfile.open ("first.txt");

    myfile << "This is my first file \n";
    myfile.close();
    return 0;
}
```

This Code will creates a file “first.txt” and insert a sentence into it in the same way we are used to do with cout<< but using the myfile stream instead

#### 11.3.1 Open a file

Open (filename, mode);

- Mode is optional parameter with a combination of the following flags

ios::in	Open for input operations.
ios::out	Open for output operations.
ios::binary	Open in binary mode. Set the initial position at the end of the file.
ios::ate	If this flag is not set to any value, the initial position is the beginning of the file.
ios::app	All output operations are performed at the end of the file, appending the content to the current content of the file. This flag can only be used in streams open for output-only operations.
ios::trunc	If the file opened for output operations already existed before, its previous content is deleted and replaced by the new one.

All these flags can be combined using the bitwise operator OR (|). For example, if we want to open the file example.bin in binary mode to add data we could do it by the following call to member function open():  
ofstream myfile;myfile.open ("example.bin", ios::out | ios::app | ios::binary);

**11.4 Que No 1 Create a text file**

Write a program that create a text file “example.txt”. Open that file and write two line in it “This is a line and This is another line” and close the file. If the file is unable to create or open then show a message “ERROR Unable to open a file”.

**11.4.1 Code**

**11.5 Que No 2 Read from the file**

Write a program that read the text from file "example.txt".

If the file is unable to open then show a message "ERROR Unable to open a file".

**11.5.1 Code**



**11.6 Que No 3 Write data through variable**

Write a program that create a text file “fdata.txt”. Open that file and write the values through variables as under. After successful writing show a message “File written successfully” .

```
char ch = 'x';  
int j = 77;  
double d = 6.02;  
string str1 = "How are u? ";  
string str2 = "Pretty Good";
```

**11.6.1 code**

## 11.7 String with Embedded blanks

Write a program that creates a file “**test.txt**” and write the following lines in that file

If we don't work hard,  
We will not be able to score high &  
might some our class fellow get flunked  
repeating the semester again

Write these lines one by one in the same format given above

### 11.7.1 Code

### Lab Evaluation Summary

Lab #	Lab Topics	Date conducted	Total Marks	Marks Obtained	Signature
1	Basic				
2	If, if-else, Nested if else				
3	Switch Statement				
4	For loop, While loop				
5	Do- while loop				
6	Function				
7	Arrays, Strings				
8	Two Dimensional Arrays				
9	Structures				
10	Pointers				
11	Files				
Total					

**Note:** It is the responsibility of every student to get this sheet signed from his/her lab engineer after every lab.