

# OBJECT ORIENTED PROGRAMMING LAB

## LAB RECORD

*Submitted by*

Palash Mishra

Submitted to: NAVALJEET SINGH SIR



**2020-2021**

Department of Computer Science & Engineering

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY,  
AB ROAD, RAGHOGARH,  
DT. GUNA-473226 MP, INDIA

# Table of Contents

<u>Lab Exercise: with topic</u>	<u>Page No.</u>
Lab Exercise1: <b>_Revisiting C</b>	<b>3</b>
Lab Exercise2: <b>Revisiting C</b>	<b>13</b>
Lab Exercise3: <b>Function Overloading</b>	<b>24</b>
Lab Exercise4: <b>Classes and Friend Functions</b>	<b>37</b>
Lab Exercise5: <b>Constructors &amp; Destructors</b>	<b>44</b>
Lab Exercise6: <b>Operator Overloading &amp; Type Conversion</b>	<b>50</b>
Lab Exercise7: <b>Inheritance</b>	<b>63</b>
Lab Exercise8: <b>Polymorphism</b>	<b>74</b>
Lab Exercise9: <b>File Handling</b>	<b>81</b>

## Lab Exercise 1: Revisiting C

### Question 1:

Write a program to round off an integer “i” to the next largest multiple of another integer “j”. For example, 256 days when rounded off to the next largest multiple divisible by a week result into 259.

```
//*****
//This program is developed by Palash Mishra (201B172)
//*****

#include<stdio.h>

int main()
{
    int i,j;

    printf("Enter the values for i and j \n");
    scanf("%d%d",&i,&j);

    if(i%j==0)
        printf("The result we want is : %d",i);
    else
        printf("The result we want is : %d",i+j-(i%j));
    return 0;
}
```

### Question 2.

A number is entered through the keyboard. The number may contain 1,2,3,4, or 5 digits. Write a program to find the number of digits in the number.

```
//*****
//This program is developed by Palash Mishra (201B172)
//*****

#include<stdio.h>

int main()
```

```

{
    int num,count=0;
    printf("Enter a number \n");
    scanf("%d",&num);
    while(num!=0)
    {
        num=num
        /10;
        count++;
    }
    if(num<=5)
        printf("The number of digits : %d",count);
    else

        printf("Your number is not valid here. try again!");
    return 0;
}

```

### Question 3.

Write a program which finds a four-digit number AABB which is a perfect square. A and B represent different digits. For example, 7744 is a four-digit perfect square number which is also satisfying the condition AABB i.e. firsttwo digits (AA=77) are same and last two digits (BB=44) are same.

```

//*****
//This program is developed by Palash Mishra (201B172)
//*****

```

```

#include
<stdio.h>
#include
<math.h>

int perfectsquare(int);
int check(int);
int main()
{
    int n;

    printf("Enter a 4-digit number in the form AABB :
    ");scanf("%d", &n);

```

```

int ch =
check(n);switch
(ch)
{
case 0:
    printf("Invalid Input");
    break;

case 1:
{
    if (perfectsquare(n))
        printf("%d satisfies the condition and is a Perfect Square.", n);
    else
        printf("%d satisfies the condition but it is not a Perfect Square.", n);
    break;
}
}
}
int perfectsquare(int x)
{
    int m = sqrt(x);
    if (m * m ==
        x) return 1;
    else
        return 0;
}
int check(int a)
{
    int r1 = a %
    100;int r2 = a /
    100;

    if (r1 % 11 == 0 && r2 % 11 == 0)

```

```
        return
1;else

        return 0;
}
```

#### Question 4.

Write a program to calculate factorial of a number N through recursion.

```
//*****
//This program is developed by Palash Mishra (201B172)
//*****
```

```
#include <stdio.h>
```

```
int fact(int
    num)
{
    if(num=
    =0)
        return 1;

    else
        if(num==1)
            return 1;

    else
        return num*fact(num-1);
}

int main()
{
    int n;

    printf("Enter the value for
    N\n");scanf("%d",&n);

    printf("The factorial for N is : %d",fact(n));
    return 0;
}
```

### Question 5.

Write a program which takes a string as input from user and returns the length of that string without using any string library functions.

```
/**
//*****
//This program is developed by Palash Mishra (201B172)
//*****
#include <stdio.h>

int main()
{
    int i=1;
    char str[100];
    printf("Enter the string\n");
    gets(str);
    for ( i = 0; str[i] != '\0'; i++); printf("The
        string length is : %d", i);return 0;
}
```

### Question 6.

Write a pointer version of the function strcat(s,t) which concatenates the string t to the end of string s.

```
/**
//*****
//This program is developed by Palash Mishra (201B172)
//*****
#include <stdio.h>
void Strcat(char *s, char *t)
{
    int i = 0, j = 0; while
    (s[i] != '\0')
    {
        i++;
    }
    while (t[j] != '\0')
    {
        s[i] = t[j];i++;
        j++;
    }
}
```

```

}
s[i]='\0';
puts(s);
}

int main()
{
char s[50], t[50]; printf("Enter
the strings \n");gets(s);
gets(t);
Strcat(s, t);

return 0;
}

```

### Question 7.

Write the function strend(s,t), which returns 1 if the string t occurs at the end of the string s, and zero otherwise.

Sample Test case1:

Input:

s="Object Oriented Programming  
using C++" t="Using C++"

Output: 1

Sample Test case2:

Input:

s="Object Oriented Programming  
using C++" t="Programming"

Output: 0

/\*\*
\*\*\*\*\*

//This program is developed by Palash Mishra (201B172)

/\*\*
\*\*\*\*\*

```

#include
<stdio.h>

```



```

#include
<string.h>

int Strend(char *s, char *t, int ls,int lt)
{
    int j=0,i=ls-lt;
    char str[50];
    while(i<=ls)
    {
        str[j]=s[i];
        i++;
        j++;
    }
    str[j]='\0';
    if(strcmpi(t,str)==0)
    return 1;
    else
    return 0;
}

int main()
{
    char s[50], t[50];
    printf("Enter the strings \n");
    gets(s);
    gets(t);
    int len_t = strlen(t),len_s=strlen(s);

    printf("%d",Strend(s, t, len_s,len_t));

    return 0;
}

```

### Advance question:

1. Write a program to find K'th smallest and K'th largest element in

unsorted array. Sample Test case1:

Input:

A[]=4, 5, 60, 70,

33, 44K=2

Output: 2nd smallest number is 5 and 2nd largest number is 60

Sample Test case2:

Input:

A[]=2, 46, 56, 68, 3, 34, 489, 457,

4545, 100K=5

Output: 5th smallest number is 56 and 5th largest number is 68

//\*\*\*\*\*

//This program is developed by Palash Mishra (201B172)

//\*\*\*\*\*

#include <stdio.h>

int Min(int \*, int);

int Max(int \*, int);

int main()

{

int n, K, temp;

printf("Enter Length of Array :  
");scanf("%d", &n);

temp = n;

int arr[n];

for (int i = 0; i < n; i++)

{

printf("Enter number %d : ", i +  
1);scanf("%d", &arr[i]);

}

printf("Enter value of K : ");

scanf("%d", &K);

int min[K], max[K];

```

for (int i = 0; i < K; i++)
{
    min[i] = Min(arr, n--);
}

n = temp;
for (int i = 0; i < K; i++)
{
    max[i] = Max(arr, n--);
}

printf("\n%d Smallest number is %d", K, min[K -
1]);printf("\n%d Largest number is %d", K,
max[K - 1]); return 0;
}

int Min(int *arr, int x)
{
    int min = arr[0], a, index;
    for (int i = 0; i < x; i++)
    {
        if (min >= arr[i])
        {
            min =
            arr[i];index
            = i;
        }
    }
    a = arr[x - 1];
    arr[x - 1] = arr[index];
    arr[index] = a;
    return min;
}

int Max(int *arr, int x)

```

```
{  
  
    int max = arr[0], a, index;  
    for (int i = 0; i < x; i++)  
    {  
        if (max <= arr[i])  
        {  
            max =  
            arr[i];index  
            = i;  
        }  
    }  
    a = arr[x - 1];  
    arr[x - 1] = arr[index];  
    arr[index] = a;  
    return max;  
}
```

## Lab Exercise 2: Revisiting C

### Question 1:

Write a function that will return the length of a character string using pointer. You are not allowed to use the strlen C library function.

```
//*****  
//This program is developed by Palash Mishra (201B172)  
//*****  
  
#include <stdio.h> int len(char *len)  
{  
    int i;  
    for ( i = 0; len[i] != '\0'; i++); return i;  
}  
  
int main()  
{  
    char str[100];  
    printf("Enter your string \n"); gets(str);  
    printf("The Length of the string : %d\n",len(str));  
  
    return 0;  
}
```

### Question 2:

Write a function that finds the minimum and the maximum value in an array of N integers. Inputs to the function are the array of integers, an integer variable containing the length of the array and pointers to integer variables that will contain the minimum and the maximum values. The function prototype is:

```
void minmax( int array[], int length, int * min, int * max);
```

Write a main function that uses this function to find and display the minimum and the

maximum values of an array of integers.

```
/**
//*****
//This program is developed by Palash Mishra (201B172)
//*****
#include <stdio.h>

void minmax(int arr[], int length, int *min, int *max)
{
    int temp;
    for (int i = 0; i < length; i++)
    {
        for (int j = 0; j < length - 1; j++)
        {
            if (arr[j] < arr[j + 1])
            {
                temp = arr[j]; arr[j] = arr[j + 1]; arr[j + 1] = temp;
            }
        }
    }
    *min = arr[length-1];
    *max = arr[0];
}

int main()
{
    int len, max, min; printf("Enter the length : \n"); scanf("%d", &len);
    int arr[len];
    for (int i = 0; i < len; i++)
    {
        scanf("%d", &arr[i]);
    }
    int *ptr = arr;
```

```

minmax(ptr, len, &min, &max);

printf("The Maximum Value is : %d \nThe Minimum Value is : %d",max,min); return 0;

}

```

### Question 3:

Write a menu driven program (using switch-case) to create a database of student names and perform the following operations using array of character pointers and dynamic memory allocation.

(A) To insert a student name

(B) To delete a name (Show Error message if zero names are there in database)

(C) To print the names

Note: your program should keep on showing above three options until user enters 'N'.

```

//*****

```

```

//This program is developed by Palash Mishra (201B172)

```

```

//*****

```

```

#include <stdio.h>

```

```

#include <stdlib.h>

```

```

#include <conio.h>

```

```

#include <string.h>

```

```

struct stud

```

```

{

```

```

char name[100];

```

```

};

```

```

int main()

```

```

{

```

```

struct stud *ptr; int sz, n;

```

```

char choice = 'Y', name_delete[50];

```

```

printf("Enter the size\n");

```

```

scanf("%d", &sz);

```

```

ptr = (struct stud *)malloc(sz * sizeof(struct stud *));

```

```

if (ptr == NULL)
{
printf("Memory Error\n");
return 0;
}
sz = 0;
while (choice != 'N')
{
printf("\t\t\t\t");
printf(" 1. Enter New Records\n\n");
printf("\t\t\t\t");
printf(" 2. Delete Record\n\n"); printf("\t\t\t\t");
printf(" 3. Print the records\n\n"); printf("\t\t\t\t");
printf("Choose Options:[1/2/3]:"); scanf("%d", &n);
switch (n)
{
case 1:
{
printf("Enter the name\n");
fflush(stdin);
gets((ptr + sz)->name);
sz++;
break;
}
case 2:
{
printf("Enter the name to be deleted\n");
fflush(stdin);
gets(name_delete);
int k = -1;

```



```

for (int i = 0; i < sz; i++)
{
if (strcmp(name_delete, (ptr + i)->name) == 0)
{
k = i; break;
}
}
if (k == -1)
{
printf("ERROR\n");
}
else
{
for (int j = k; j < sz - 1; j++)
{
strcpy((ptr + j)->name, (ptr + j + 1)->name);
}
sz--;
ptr = (struct stud *)realloc(ptr, sz * sizeof(struct stud *));
}
break;
}
case 3:
{
printf("Names of the students are\n");
if (sz == 0)
printf("ERROR\n"); else
{
for (int i = 0; i < sz; i++)
printf("%s\n", (ptr + i)->name);
}
}

```

```

}
break;
}
default:
printf("ERROR-Invalid Input");
return 0;
}
printf("Enter 'Y' to continue or 'N' to stop\n");
fflush(stdin);
scanf("%c", &choice);
}
free(ptr); ptr = NULL;
return 0;
}

```

#### Question 4:

Write a program to generate random numbers in given range [m, n].

Test case :

Input: m=10, n=50

Output: 34

```

//*****
//This program is developed by Palash Mishra (201B172)
//*****
#include <stdio.h> #include <stdlib.h> #include <time.h>

void printRandoms(int up , int down ,
int nums )
{
int i;

```

```

for (i = 0; i < nums; i++)
{
    int num = (rand() %
    (up - down + 1)) + down;
    printf("%d ", num);
}
}

```

```

int main()
{
    int m,n,num_s;

    printf("Enter the upper bound : \n"); scanf("%d",&m);
    printf("Enter the lower bound : \n"); scanf("%d",&n);
    printf("Enter value that how many random numbers you want : \n"); scanf("%d",&num_s);
    srand(time(0));
    printRandoms(m,n,num_s); return 0;
}

```

#### Question 5:

An automobile company has serial number for engine parts starting from AA0 to FF9.

The other characteristics of parts to be specified in a structure are: Year of manufacturing, material (Steel, Aluminum, Iron etc.) and quantity manufactured.

Write a program to include following tasks:

- (a) Specify a C structure that includes four members viz. serial numbers, year of manufacturing, material type, and quantity. Declare an array of structure of size 60.
- (b) Automatically initialize the values of structure members as follows :
  - i. Generate serial numbers such that first part has serial number AA0 and 60th part has serial number FF9.
  - ii. Randomly generate year of manufacturing in the range [1990-2018]

- iii. Randomly initialize material type from three choices ie. Steel, Aluminum, Iron
  - iv. Randomly initialize quantity in the range [1-1000]
- (c) Display the information of the parts with serial numbers between any given range such as [BB1,CC6].

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include <time.h>
```

```
char *random_ser();
```

```
char str[60][3];
```

```
char ch = 'A', ch2 = '0'; int n = 0, k;
```

```
typedef struct company
```

```
{ char ser_num[4]; int year;
```

```
char material[20]; int qty;
```

```
} com; int main()
```

```
{ int q; srand(time(0)); int num;
```

```
com arr[60];
```

```
for (int i = 0; i < 60; i++)
```

```
{
```

```
q = rand() % 3; strcpy(arr[i].ser_num, random_ser());
```

```
arr[i].year = (rand() % (2018 - 1990 + 1)) + 1990; if (q == 0)
```

```
strcpy(arr[i].material, "Steel "); else if (q == 1)
```

```
strcpy(arr[i].material, "Aluminium"); else
```

```
strcpy(arr[i].material, "Iron "); arr[i].qty = (rand() % (1000 - 1 + 1)) + 1;
```

```
}
```

```

printf("
(s.no.)  Serial number Year  Quantity  Material
\n"); printf("
\n"); for (int j = 0; j < 60; j++)
{
printf("( %d)  %s  %d  %d  %s  \n", j + 1, arr[j].ser_num, arr[j].year, arr[j].qty, arr[j].material);
}

printf("\n"); char ser1[3], ser2[3];
printf("Enter the range for the serial number\n"); gets(ser1);
gets(ser2); int s, e;
for (k = 0; k < 60; k++)
{
if (arr[k].ser_num[1] == ser1[1] && arr[k].ser_num[2] == ser1[2])
{
s = k;
}
if (arr[k].ser_num[1] == ser2[1] && arr[k].ser_num[2] == ser2[2])
{
e = k; break;
}
}
for (int z = s; z <= e; z++)
{
printf("( %d)  %s  %d  %d  %s  \n", z + 1, arr[z].ser_num, arr[z].year, arr[z].qty,
arr[z].material);
}
return 0;
}

char *random_ser()
{
char *ptr; int i, j;
for (j = 0; j < 3; j++)
{

```

```
str[i][j] = ch2; if (j < 2)
```

```
{
```

```
str[i][j] = ch;
```

```
}
```

```
}
```

```
ch2++; n++;
```

```
while (n > 9)
```

```
{
```

```
n -= 10;
```

```
ch++;
```

```
ch2 = '0';
```

```
}
```

```
ptr = str; return ptr;
```

```
}
```

### Advance Question 1:

Given an array A of size N-1 and given that there are numbers from 1 to N with one element missing; Write program to find the missing number.

Test case 1: Given array: 1 2 3 5; missing element is 4.

Test case 2: Given array: 1 2 3 4 5 6 7 8 10; missing element is 9.

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int N, t_case, sum = 0;
```

```
printf("Enter the number of test cases : \n"); scanf("%d", &t_case);
```

```
for (int i = 1; i <= t_case; i++)
{
printf("Enter the size \n"); scanf("%d", &N);
int arr[N];

for (int j = 0; j < N; j++)
{
scanf("%d", &arr[j]);
}
sum = ((N + 1) * (N + 2)) / 2;
for (int k = 0; k < N; k++)
{
printf("%d ", arr[k]); sum = sum - arr[k];
}
printf("Missing element is : %d", sum); printf("\n");
}

return 0;}
```

### Lab Exercise-3 (Reference variables, functionoverloading, static variables)

#### Question 1:

Write C++ Program to swap two variable using reference variables.

```
/**
//This program is developed by Palash Mishra (201B172)
**/

#include <iostream>
using namespace std;
int main()
{
int var1, var2, temp;
int &ref1 = var1, &ref2 = var2;
cout << "Enter the values for var1 and var2" << endl; cin >> var1 >> var2;
cout << "values before swapping var 1=" << ref1 << " and var2 =" << ref2 << endl; temp = ref1;
ref1 = ref2; ref2 = temp;
cout << "values After swapping var1 =" << ref1 << " and var2 =" << ref2 << endl;

return 0;
}
```

#### Question 2:

Write a function that finds the minimum and the maximum value in an array of N integers. Inputs to the function are the array of integers, an integer variable containing the length of the array and references to integer variables that will contain the minimum and the maximum values. The function prototype is:

```
void minmax (int array[], int length, int& min, int & max);
```

```
/**
//This program is developed by Palash Mishra (201B172)
```



```
//*****
```

```
#include <iostream>

using namespace std;

void minmax(int arr[], int length, int &min, int &max)
{
    int temp;
    for (int i = 0; i < length; i++)
    {
        for (int j = 0; j < length - 1; j++)
        {
            if (arr[j] < arr[j + 1])
            {
                temp = arr[j]; arr[j] = arr[j + 1]; arr[j + 1] = temp;
            }
        }
    }
    min = arr[length - 1];

    max = arr[0];

    cout<<"The Maximum Value is : "<<max<<endl<<"The Minimum Value is : "<<min;

}

int main()

{
```

```
int len, maxi, mini;
```

```
cout<<"Enter the length : ";
```

```
cin>>len;
```

```
int arr[len];
```

```
for (int i = 0; i < len; i++)
```

```
{
```

```
cin>>arr[i];
```

```
}
```

```
int *ptr = arr;
```

```
minmax(ptr, len, mini, maxi);
```

```
return 0;
```

```
}
```

### Question 3:

Create a four-function calculator for fractions. Here are the formulas for the four arithmetic operations applied to fractions:

Addition:  $a/b + c/d = (a*d + b*c) / (b*d)$

Subtraction:  $a/b - c/d = (a*d - b*c) / (b*d)$

Multiplication:  $a/b * c/d = (a*c) / (b*d)$

Division:  $a/b \div c/d = (a*d) / (b*c)$

The user should type the first fraction (two values a and b), an operator, and a second fraction (two values c and d). The program should then display the results in fraction ie. (Numerator/ denominator).

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
float a, b, c, d;
```

```
char res;
```

```
cout << "Enter the values for the a b " << endl; cin >> a >> b;
```

```
cout << "Enter an operator " << endl; cin >> res;
```

```
cout << "Enter the values for the c d " << endl; cin >> c >> d;
```

```
cout<<"Your result is : "; if (res == '+')
```

```
{
```

```
cout << (a * d + b * c) << "/" << (b * d);
```

```
}
```

```
else if (res == '-')
```

```
{
```

```
cout << ((a * d) - (b * c)) << "/" << (b * d);
```

```
}
```

```
else if (res == '*')
```

```
{
```

```
cout << (a * c) << "/" << (b * d);
```

```
}
```

```

else if (res == '/')
{
cout << (a * d)<<"/"<<(b * c);
}
else
{
cout << "Please enter the correct response" << endl;
}

return 0;
}

```

#### Question 4:

Create a class rectangle with attributes length and width. Provide member functions that calculate the perimeter and area of the rectangle. Provide member functions to get the values from users and display the values of member variables. Write a program to test the class.

```

//*****
//This program is developed by Palash Mishra (201B172)
//*****

```

```

#include<iostream>

using namespace std;

class rectangle
{
private:
float length ,width; float a , p;
public:
void get_value()
{
cout<<"ENTER length and width : "; cin>>length>>width;

```

```

}

void display()
{
cout<<"\n"<<"THE VALUES OF LENGTH AND WIDTH ARE : "<<length<<" "<<
width;
}

float perimeter()
{
p=2*(length+width); return p;
}

float area()
{
a=length * width ; return a;
}

};

int main()
{
rectangle r; r.get_value(); r.display();
cout<<"\n"<<"The perimeter of rectangle is ="<<r.perimeter()<<"\nThe area of rectangle is ="<<r.area();
return 0;}

```

#### Question 5:

Write a function that accepts two arguments: a string name of a movie and an integer running time in minutes. Provide a default value for the minutes so that if you call the function without an integer argument, the minutes default to 90. Write a main() function that proves you can call the function with a string argument alone as well as with a string and an integer.

```

//*****

```

```

//This program is developed by Palash Mishra (201B172)

```

```

//*****

```

```

#include <iostream>

```

```

using namespace std;

void movie(char name[], int = 90)

{

}

int main()

{

char mo_name[20]; int mo_time;

movie(mo_name, mo_time); movie(mo_name);

}

```

#### Question 6:

Create a class named Shirt that has the public data members collarSize and sleeveLength.

Create a class named Pants that has the public data members waistSize and inSeam. Write a program that declares one object of each type Shirt and Pants and assigns values to the objects' data fields. Write two overloaded functions named displayClothingFacts(). One version of the function takes a Shirt object as an argument; the other version takes a Pants object. Each version displays the facts about the piece of clothing. Your main() function should demonstrate that you can call displayClothingFacts() with either type of clothing.

```

//*****

//This program is developed by Palash Mishra (201B172)

//*****

```

```

#include <iostream>

using namespace std;

class Shirt

{

public:

float collarSize, sleeveLength;

};

class Pants

{

```

```

public:
float inSeam, waistSize;

};

void displayClothingFacts(Shirt s)
{
cout << "\n"
<< "COLLARSIZE : " << s.collarsize << "\nSLEEVE LENGTH : " << s.sleeveLength;
}

void displayClothingFacts(Pants p)
{
cout << "\n"
<< "INSEAM : " << p.inSeam << "\nwaistsize : " << p.waistSize;
}

int main()
{
Shirt s1; Pants p1;
s1.collarsize = 15.5;
s1.sleeveLength = 45.98;
p1.inSeam = 60.5;
p1.waistSize = 55.4;
displayClothingFacts(s1); displayClothingFacts(p1); return 0;
}

```

#### Question 7:

Define a class named Movie. Include private fields for the title, year, and name of the director. Include three public functions with the prototypes void Movie::setTitle(string); , void Movie::setYear(int); and void Movie::setDirector(string);. Include another function that displays all the information about a Movie. Write a main() function that declares a movie object named myFavoriteMovie. Set and display the object's fields.

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <iostream>
```

```
#include <string.h>
```

```
using namespace std;
```

```
class Movie
```

```
{
```

```
char title[20]; int year;
```

```
char name[20];
```

```
public:
```

```
void setTitle(char t[])
```

```
{
```

```
strcpy(title, t);
```

```
}
```

```
void setYear(int y)
```

```
{
```

```
year = y;
```

```
}
```

```
void setDirector(char n[])
```

```
{
```

```
strcpy(name,n);
```

```
}
```

```
void display()
```

```
{
```

```
cout << title <<endl << year <<endl << name <<endl;
```

```
}
```



```

};

int main()

{
    Movie myFavoriteMovie;

    myFavoriteMovie.setTitle("Jungle Book");
    myFavoriteMovie.setYear(2016);

    myFavoriteMovie.setDirector("Jon Favreau"); myFavoriteMovie.display();
    return 0;
}

```

#### Question 8:

Write a class definition for an order class for a nightclub that contains a table number, a server's name, and the number of patrons at the table. Include a private static data member for the table minimum charge, which is \$4.75. Write a main() function that declares no object of order class type, but uses a static member function to display the table minimum charge.

```

//*****

//This program is developed by Palash Mishra (201B172)

//*****

#include <iostream>
using namespace std;
class order
{
    static float min_charge;

public:
    int tb_num, num_patrons; char ser_name[20];

    static void display()

```

```

{
cout <<"Table Minimum Charge : "<< min_charge;

}

};

float order::min_charge=4.75; int main()

{

order::display();

}

```

## Advanced Problems:

### Question 9:

Write a c++ program to find the highest occurring digit in prime numbers in a given range.

Given a range L to R, the task is to find the highest occurring digit in prime numbers lie between L and R (both inclusive). If multiple digits have same highest frequency print the largest of them. If no prime number occurs between L and R, output -1.

Examples:

Input : L = 1 and R = 20.

Output : 1

Prime number between 1 and 20 are 2, 3, 5, 7, 11, 13, 17, 19.

1 occur maximum i.e 5 times among 0 to 9.

```

//*****

```

```

//This program is developed by Palash Mishra (201B172)

```

```

//*****

```

```

#include <iostream>

```

```

using namespace std;

```

```

int arr[10], temp;

```

```

void prime(int);

```

```

int main(void)

```

```

{

```

```

int a, b, i, max, no;

```

```

cout<<"enter the two no:"<<endl; cin>>a>>b;

for (i = a; i <= b; i++) prime(i);

if (temp == 0)
{
cout<<"NO ANY PRIME NO"<<endl;
}
else
{
max = arr[0]; no = 0;
for (i = 1; i <= 9; i++)
{
if (arr[i] >= max)
{
max = arr[i]; no = i;
}
}
cout<<"Output:"<<no;

}

return 0;
}

void prime(int m)
{
int a;
for (a = 2; a <= m / 2; a++)
{
if (m % a == 0) return;
}
}

```

```
temp = 1; while (m > 0)
{
a = m % 10;
m = m / 10; arr[a] += 1;
}
}
```

## Lab Exercise-4 (Classes and Friend Functions)

### Question 1:

1. The annual examination results of 100 students are tabulated as follows:

---

Roll No.	Subject 1	Subject 2	Subject 3
....			
....			
....			

Write a program to read the data and determine the following:

- Total marks obtained by each student.
- The highest marks in each subject and the Roll No. of the students who secured it.
- The student who obtained the highest total marks.

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <iostream>
```

```
using namespace std;
```

```
class student
```

```
{
```

```
public:
```

```
int roll_no;
```

```
float sub1, sub2, sub3; float total;
```

```
void input()
```

```
{
```

```
cout << "Enter the roll number,sub1,sub2,subj3" << endl; cin >> roll_no >> sub1 >> sub2 >> sub3;
```

```
total = sub1 + sub2 + sub3;
```

```

}
};

int main()
{

student st[5];
for (int i = 0; i < 5; i++)
{
st[i].input();
}

int high = 0;
for (int i = 0; i < 5; i++)
{

if (st[i].sub1 > st[high].sub1)
{
high = i;
}
}

cout << "Student who got highest in subject 1" << endl;
cout<<"ROLL NO . : "<<st[high].roll_no<<"\t"<<"Marks : " <<st[high].sub1<<endl; high = 0;
for (int i = 0; i < 5; i++)
{

if (st[i].sub2 > st[high].sub2)
{
high = i;
}
}

```

```

}

cout << "Student who got highest in subject 2" << endl;

cout<<"ROLL NO . : "<<st[high].roll_no<<'\t'<<"Marks : " <<st[high].sub2<<endl; high = 0;

for (int i = 0; i < 5; i++)

{

if (st[i].sub3 > st[high].sub3)

{

high = i;

}

}

cout << "Student who got highest in subject 3" << endl;

cout<<"ROLL NO . : "<<st[high].roll_no<<'\t'<<"Marks : " <<st[high].sub3<<endl; high = 0;

for (int i = 0; i < 5; i++)

{

if (st[i].total > st[high].total)

{

high = i;

}

}

cout<<"The student who obtained the highest total marks"<<endl;

cout<<"ROLL NO . : "<<st[high].roll_no<<'\t'<<"Marks : " <<st[high].total<<endl;


return 0;

}

```

### Question 2:

Define a class to represent a bank account, including the following data members:

§ Name of the depositors

§ Account number

§ Type of account

§ Balance amount in the account

and member functions:

§ To assign initial values

§ To deposit an amount

§ To withdraw an amount after checking the balance

§ To display the name and balance

Write a main program to test the program.

```
//************************************************************************
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//************************************************************************
```

```
#include <iostream>
```

```
#include<string.h>
```

```
using namespace std;
```

```
class account
```

```
{
```

```
char name[20];
```

```
long long int acc_num; char type[20];
```

```
float bal_amt;
```

```
public:
```

```
void assign(); void credit(); void debit(); void detail();
```

```
};
```

```
void account ::assign()
```

```
{
```

```
strcpy(name, "Palash"); acc_num = 7999268563; strcpy(type, "CURRENT"); bal_amt = 15000;
```

```
}
```



```
void account ::credit()
{
float amount;
cout << "\nEnter the amount to be Credited\n"; cin >> amount;
bal_amt += amount;
}

void account ::debit()
{
float amount;
cout << "\n Current Balance : " << bal_amt; cout << "\nEnter the amount to be Debited\n"; cin >> amount;
if (bal_amt - amount >= 0) bal_amt -= amount;

else
}

cout << "\nNOT ENOUGH BALANCE \n";

void account ::detail()
{cout << "\nNAME : " << name;
cout << "\nBALANCE : " << bal_amt;
}

int main()
{
account acc; acc.assign();
acc.credit();
acc.debit();
acc.detail();

return 0;
```

```
}
```

Question 3:

Create two classes DM and DB which store the value of the distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB.

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include<conio.h>
```

```
#include<iostream>
```

```
using namespace std;
```

```
class DM
```

```
{    public: float meter;
```

```
void getdata()
```

```
{cout<< "Enter Distance in Meters and Centimeters : " ; cin>> meter ;
```

```
}
```

```
friend void _add();
```

```
};
```

```
class DB
```

```
{    public: float inch;
```

```
void getdata()
```

```
{cout<< "Enter Distance in Feets and Inches : " ; cin>> inch ;
```

```
}
```

```
friend void _add();
```

```
};
```

```
void _add()
```

```
{    float temp,res; temp=dm.meter*39.37; res=temp+db.inch;
```

```
cout<< " Total inches = " << res <<endl; temp=db.inch/39.37;  
res=temp+dm.meter;  
cout<< " Total meters = " << res ;  
}  
main()  
{      DM dm; DB db;  
dm.getdata();  
db.getdata();  
_add();  
getch(); return 0;  
  
}
```

## Lab Exercise 5(Constructors and Destructors)

### Question 1:

Write a program to perform addition of two complex numbers using constructor overloading. The first constructor which takes no argument is used to create objects which are not initialized, second which takes one argument is used to initialize real and imaginary parts to equal values and third which takes two argument is used to initialize real and imaginary to two different values. Include sum member function that takes two objects and performs the addition of these two objects. Write display function to display the object in complex form i.e.  $2+i6$  . Write main function to test your program.

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <iostream>
```

```
using namespace std;
```

```
class complex
```

```
{
```

```
public:
```

```
int x, y; complex()
```

```
{
```

```
x = 0;
```

```
y = 0;
```

```
}
```

```
complex(int t)
```

```
{
```

```
x = t; y = t;
```

```
cout << "The sum of your complex number is : " << x << "+" << y << "i"
```

```
<< endl;
```

```

}

complex(int r, int i)
{
x = r; y = i;
cout << x << " " << y << endl;
}

void sum(complex &ob1, complex &ob2)
{

x = ob1.x + ob2.x; y = ob1.y + ob2.y;
}

void display()
{

cout << "The sum of your complex number is : " << x << "+" << y << "i"
<< endl;
}
};

int main()
{
int real, img, value; complex fi;
cout << "Enter a value to make real and img part equal :\n"; cin >> value;
complex sec(value);
cout << "NUMBER 1 : \n";
cout << "Enter your number real and img part :\n"; cin >> real >> img;
complex thi_1(real, img); cout << "NUMBER 2 : \n";
cout << "Enter your number real and img part :\n"; cin >> real >> img;
complex thi_2(real, img); thi_1.sum(thi_1, thi_2); thi_1.display();

```

```
return 0;
```

```
}
```

### Question 2:

A point in a two-dimensional plane having coordinate as (x,y), can be represented by a class whose private data members are x and y. Write the constructor and destructor functions of the class. The constructor should initialise (x,y) by passing parameters values. Now, a rectangle can be represented by the top-left and bottom-right vertices.

Define a class say 'Rectangle' whose private data members are top-left and bottom-right vertices. Write the parameterised constructor function of the class 'Rectangle'. Also, write the destructor function. Finally, write a program to show the order in which different constructors and destructors are called.

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <iostream>
```

```
using namespace std;
```

```
class plane
```

```
{
```

```
int x, y;
```

```
public:
```

```
plane(int x, int y)
```

```
{
```

```
cout << "Constructor for plane class called \n";
```

```
}
```

```
~plane()
```

```
{
```

```
cout << "Destructor for plane class called \n";
```

```
}
```

```
};
```

```

class rectangle
{
int top_l, bottom_r;

public:
rectangle(int a, int b)
{
cout << "Constructor for rectangle class called \n";
}
~rectangle()
{
cout << "Destructor for rectangle class called \n";
}
};

int main()
{
class plane pl(2, 3);
class rectangle rc(5,9);
return 0;
}

```

### Question 3:

Write a program to show that, the constructor and destructor functions of a globally declared object are the first and last functions, respectively to be called in a program.

```

//*****
//This program is developed by Palash Mishra (201B172)
//*****

#include <iostream>

```

```

using namespace std;

class oop
{
int x;

public:
oop(int y = 0) : x(y)
{
cout << "Constructor Calls" << x << "() constructor " << endl;
}
~oop()
{
cout << "Destructor Calls" << x << "() destructor " << endl;
}
};

int main()
{
oop m;
}

```

#### Question 4:

Write a program to show that constructors follow the property of function overloading as well as default parameter. Also, show that in case of constructor also, default parameter may create problem in implementing function overloading.

/\*Write a program to show that constructors follow the property of function overloading as well as default parameter. Also, show that in case of constructor also, default parameter may create problem in implementing function overloading.\*/

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```



```

#include <iostream>

using namespace std; class cons
{
int x;
public:
cons()
{ cout << "Constructor 1 Called" << endl; }
cons(int y)
{
cout << "Constructor 2 Called" << endl;
}
cons(char c)
{
cout << "Constructor 3 Called" << endl;
}
cons(double y)
{
cout << "Constructor 4 Called" << endl;
}
cons(int y=0)
{
cout << "Constructor 1 Called" << endl;
}};

int main()
{
cons m; cons n(4); cons o('p');
cons p(5.8);
cons q(5);
}

```

## Lab Exercise 6(Operator Overloading and Type Conversion)

### Question 1:

Create two classes polar and rectangle. Polar class has two data members radius and angle and rectangle has two data members x and y. Write constructors and member functions to get input from user and display the data members in both the classes. Write a program using the concept of type conversion to convert an object of class polar to object of rectangle and vice versa. Also write a function which computes distance between two points represented either in polar or rectangular coordinates.

Hint: formulas to convert polar coordinates to rectangular coordinates:

$$x = r \cos q, y = r \sin q$$

formulas to convert rectangular coordinates to polar coordinates:

$$r = \sqrt{x^2 + y^2}, q = \tan^{-1}(y/x)$$

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
class Polar
```

```
{
```

```
public:
```

```
double radius, angle;
```

```
Polar()
```

```
{
```

```
setRadius();
```

```
setAngle();
```

```
display();
```

```
}
```

```
void setRadius()
```

```
{
cout << "Enter the radius: ";
cin >> radius;
}

void setAngle()
{
cout << "Enter the angle in radians: ";
cin >> angle;

}

void setRadius(double rad)
{
this->radius = rad;
}

void setAngle(double angle)
{
this->angle = angle;
}

void display()
{
cout << "Radius = " << radius << ", Angle = " << angle << endl;
}
};

class Rectangle
{
public:
double x, y;
Rectangle()
{
setX();
```

```

setY();
display();
}
void setX()
{
cout << "Enter x: ";
cin >> x;
}
void setY()
{
cout << "Enter y: ";
cin >> y;
}
void setX(double x)
{
this->x = x;
}
void setY(double y)
{
this->y = y;
}
void display()
{
cout << "x = " << x << ", y = " << y << endl;
}
};
Rectangle convertPolarToRectangle(Polar p)
{
Rectangle r;

```

```

double x = p.radius * cos(p.angle);
double y = p.radius * sin(p.angle);
r.setX(x);
r.setY(y);
r.display();
return r;
}

Polar convertRectangleToPolar(Rectangle r)
{
Polar p;
double radius = sqrt(r.x * r.x + r.y * r.y);
double angle = atan(r.y / r.x);
p.setAngle(angle);
p.setRadius(radius);
p.display();
return p;
}

double distance(Rectangle r1, Rectangle r2)
{
double d = sqrt((r1.x - r2.x) * (r1.x - r2.x) + (r1.y - r2.y) * (r1.y - r2.y));
return d;
}

double distance(Polar p1, Polar p2)
{
double d = sqrt(p1.radius * p1.radius + p2.radius * p2.radius -
2 * p1.radius * p2.radius * cos(p1.radius - p2.angle));
return d;
}

int main()
{

```

```

Polar p;
Rectangle r;
cout<<"\n_____
_____ \n";
convertRectangleToPolar(r);
convertPolarToRectangle(p);

cout<<"\n_____
_____ \n";
cout <<"Distance between two points in rec. coordinates : "<< distance(r, convertPol
arToRectangle(p)) << endl;
cout <<"Distance between two points in pol. coordinates : "<< distance(p, convertRec
tangleToPolar(r));
cout<<"\n_____
_____ \n";

return 0;
}

```

### Question 2:

Design a class Distance that includes following data members: feet, inches. It has the following member function:-

- Constructor, that initializes the distance to 0,0 by default.
- Give a check so that the inches part is always less than 12.0.
- Display function
- Overloaded – operator to subtract 2 distances
- Overloaded + operator to add 2 distances
- Overload += and -= operator
- Overload > and < operators to compare two distances

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <iostream>

using namespace std;

class Distance
{
    int feet, inch;

public:
    Distance(int f = 0, int i = 0)
    {
        feet = f;
        inch = i;
    }

    Distance operator-(Distance const &obj)
    {
        Distance res;
        res.inch = inch - obj.inch;
        if (res.inch > 12)
        {
            res.feet = feet - obj.feet + (res.inch / 12);
            res.inch = (res.inch % 12);
        }
        else
            res.feet = feet - obj.feet;
        return res;
    }

    void print1() { cout << feet << " feet " << inch << " inch " << endl; }

    Distance operator+(Distance const &obj)
    {
```

```

Distance res;

res.inch = inch + obj.inch;

if (res.inch > 12)
{
res.feet = feet + obj.feet + (res.inch / 12);
res.inch = (res.inch % 12);
}

else
res.feet = feet + obj.feet;

return res;
}

void print2() { cout << feet << " feet " << inch << " inch " << endl; }

Distance operator+=(Distance const &obj)
{
feet += obj.feet;
inch += obj.inch;
}

void print3() { cout << feet << " feet " << inch << " inch " << endl; }

Distance operator-=(Distance const &obj)
{
feet -= obj.feet;
inch -= obj.inch;
}

void print4() { cout << feet << " feet " << inch << " inch " << endl; }

bool operator>(const Distance &obj)
{
if (feet > obj.feet)
{
return true;
}
}

```



```
if (feet == obj.feet && inch > obj.inch)
```

```
{
```

```
return true;
```

```
}
```

```
return false;
```

```
}
```

```
bool operator<(const Distance &obj)
```

```
{
```

```
if (feet < obj.feet)
```

```
{
```

```
return true;
```

```
}
```

```
if (feet == obj.feet && inch < obj.inch)
```

```
{
```

```
return true;
```

```
}
```

```
return false;
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
Distance d1(10, 50), d2(4, 4);
```

```
Distance d3 = d1;
```

```
Distance D_1 = d1 - d2;
```

```
D_1.print1();
```

```
Distance D_2 = d1 + d2;
```

```
D_2.print2();
```

```
d1 += d2;
```

```
d1.print3();
```

```

d3 -= d2;
d3.print4();
if (d1 > d2)
cout << "Object 1 (d1) is the greatest \n";
else
cout << "Object 2 (d2) is the greatest \n";
if (d1 < d2)
cout << "Object 1 (d1) is the smallest \n";
else
cout << "Object 2 (d2) is the smallest \n";
}

```

### Question 3:

Create a class rational for performing arithmetic with fractions. Use an integer variable to represent the private data of the class-the numerator and denominator. Provide a member function to get input from the user. This function should also check that denominator entered is not 0, if it is zero print invalid input. Provide a function to display the values. Overload +, -, \*, / operators to add, subtract, multiply and divide the objects of this class.

```

//*****
//This program is developed by Palash Mishra (201B172)
//*****

#include <iostream>
using namespace std;
class Rational
{
double num, den;
public:
void input()
{
cout << "Enter the data \n";
cin >> num >> den;

```

```

if (den == 0)
{
cout << "Invalid Input\n";
}
}

void display()
{
cout << num << "/" << den << endl;
}

Rational operator+(Rational const &obj)
{
Rational res;
if (den == obj.den)
{
res.num = num + obj.num;
res.den = den;
}
else
{
res.num = (num * obj.den) + (den * obj.num);
res.den = den * obj.den;
}

cout << "Your solution :-
" << res.num << " / " << res.den << " = " << res.num / res.den << endl;
return res;
}

Rational operator-(Rational const &obj)
{
Rational res;

```

```

if (den == obj.den)
{
res.num = num - obj.num;
res.den = den;
}
else
{
res.num = (num * obj.den) - (den * obj.num);
res.den = den * obj.den;
}
cout << "Your solution :-
" << res.num << " / " << res.den << " = " << res.num / res.den << endl;
return res;
}

Rational operator*(Rational const &obj)
{
Rational res;
res.num = num*obj.num;
res.den = den * obj.den;
cout << "Your solution :-
" << res.num << " / " << res.den << " = " << res.num / res.den << endl;
return res;
}

Rational operator/(Rational const &obj)
{
Rational res;
res.num = num*obj.den;
res.den = den * obj.num;
cout << "Your solution :-
" << res.num << " / " << res.den << " = " << res.num / res.den << endl;

```

```

return res;

}

};

int main()
{
Rational R1, R2;
R1.input();
R1.display();
R2.input();
R2.display();
Rational R3 = R1 + R2;
Rational R4 = R1 - R2;
Rational R5 = R1 * R2;
Rational R6 = R1 / R2;
}

```

#### Question 4:

Include a function that adds two strings to make a third string. Write a program to do the following tasks:

- i. Create uninitialized string objects
- ii. Creates the objects with string constants.
- iii. Concatenates two strings properly.
- iv. Displays a desired string object

```

//*****

//This program is developed by Palash Mishra (201B172)

//*****

#include <iostream>
using namespace std;
string concatStrings(string str1, string str2);
int main()
{

```

```
string str1, str2, str3;
cout << "Enter String1:";
cin >> str1;
cout << "Enter String2:";
cin >> str2;
str3 = concatStrings(str1, str2);
cout << "String3:" << str3 << endl;
return 0;
}
string concatStrings(string str1, string str2)
{
return str1 + " " + str2;
}
```

## Lab Exercise 7 (Inheritance)

### Question 1:

Write a program with a mother class and a derived daughter class. Both of them should have a method void display () that prints a message (different for mother and daughter). In the main function declare an object of class daughter and call the display() method on it. Also suitably invoke the display() function of mother class using this object of class daughter.

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <iostream>
```

```
using namespace std;
```

```
class Mother
```

```
{
```

```
public:
```

```
void display()
```

```
{
```

```
cout << "I Belong's to Mother's Class\n";
```

```
}
```

```
};
```

```
class Daughter : public Mother
```

```
{
```

```
public:
```

```
void display()
```

```
{
```

```
cout << "I Belong's to Daughter's Class\n";
```

```
}
```

```
};
```

```
int main()
{
Daughter D; D.display(); D.Mother::display(); return 0;
}
```

### Question 2:

Consider an example of declaring the examination result. Design three classes: Student, Exam, and Result. The Student class has data members representing roll number, name. Create the class Exam by inheriting Student class. The Exam class adds fields (data members) representing the marks scored in six subjects. Derive the Result from the Exam class, and it has its own fields such as total\_marks. Write an interactive program to model this relationship.

```

//*****
//This program is developed by Palash Mishra (201B172)
//*****

#include <iostream>
using namespace std;

class Student
{
int roll_no; char name[20];

public:
void getdata()
{
cout << "Enter Name \n"; cin >> name;
cout << "\nEnter Roll Number : \n"; cin >> roll_no;
}
void show()
{
cout << "roll_no : " << roll_no << "\t"
<< "Name : " << name << endl;
}
}

```



```

}
};

class Exam : public Student
{
public:
int Eng, phy, chem, bio, math, hindi; void input()
{
cout << "Enter the marks of students in the six subjects\n"; cout << "Eng, phy, chem, bio, math, hindi;\n";
cin >> Eng >> phy >> chem >> bio >> math >> hindi;
}
};

class Result : public Exam
{
int total_marks;

public:
int total()
{

total_marks = Eng + phy + chem + bio + math + hindi; return total_marks;
}

void display()
{
cout << "\nThe total of marks : " << total();
}
};

int main()
{
Result R; R.getdata();
R.show();
}

```

```
R.input();  
R.display();  
}
```

Question 3:

There is a class student, that stores name of school or university from which he is enrolled and name of highest degree he has obtained so far. It has the function to get and display the members. Design a class Employee with name and employee number. Derive Manager, Scientist and Laborer classes from Employee class. The manager class has extra attribute title (string type) and dues (float type). The scientist class has extra attributes number of publications. The Laborer class has nothing extra. The classes have necessary functions for set and display the information. The manager and scientist are students of a university also. Use inheritance. Test your program by creating objects of type manager, scientists and laborer.

```
//*****  
//This program is developed by Palash Mishra (201B172)  
//*****  
  
#include <iostream>  
using namespace std;  
  
class student  
{  
    char n_of_s[40]; char degree[30];  
  
public:  
    void gdata()  
    {  
        cout << "\nEnter name of school/university : "; cin >> n_of_s;  
        cout << "\nEnter name of highest degree : "; cin >> degree;  
    }  
    void sdata()  
    {
```

```
cout << "University : " << n_of_s << endl; cout << "Highest degree: " << degree << endl;
}
};
```

```
class employee
```

```
{
```

```
int no;
```

```
char name[20];
```

```
public:
```

```
void getdata()
```

```
{
```

```
cout << "Enter Number : "; cin >> no;
```

```
cout << "\nEnter Name : "; cin >> name;
```

```
cout << endl;
```

```
}
```

```
void showData()
```

```
{
```

```
cout << "employee no.: " << no << endl;
```

```
cout << "name of employee : " << name << endl;
```

```
}
```

```
};
```

```
class manager : public employee, public student
```

```
{
```

```
char title[40]; float dues;
```

```
public:
```

```
void get()
```

```
{
```

```
cout << "\nEnter title : "; cin >> title;
```

```

cout << "\nEnter dues : "; cin >> dues;

cout << endl;

}

void show()

{

cout << "\ntitle : " << title; cout << "\ndues : " << dues;

}

};

class scientist : public employee, public student

{

int n_of_publ;

public:

void enter()

{

cout << "\nEnter Number of Publications: "; cin >> n_of_publ;

}

void display()

{

cout << "\nNo. of publications : " << n_of_publ;

}};

class laborer : public employee

{

};

int main()

{

manager mn; mn.gdata();

mn.sdata();

mn.getdata(); mn.showData(); mn.get();

```

```

mn.show(); scientist st; st.gdata();
st.sdata();
st.getdata(); st.showData(); st.enter();
st.display(); laborer lb; lb.getdata(); lb.showData(); return 0;
}

```

#### Question 4:

An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in Fig.1. The figure also shows the minimum information required for each class. Specify all the classes and define methods to create the database and retrieve individual information as and when required.

```

//*****
//This program is developed by Palash Mishra (201B172)
//*****

#include <iostream>
using namespace std;
class staff
{
protected:
int code; string name;

public:
void sinput()
{
cout << "\nEnter code : "; cin >> code;
cout << "\nEnter name : "; cin >> name;
}
void idisplay()
{

```

```

cout
<< "\nCode : " << code; cout << "\nName : " << name;
}
};

class teacher : public staff
{
protected:
string sub; string pub;

public:
void tinput()
{
sinput();
cout << "\nEnter subject : "; cin >> sub;
cout<< "\nEnter publication : "; cin >> pub;
}
void tdisplay()
{
idisplay();
cout << "\nSubject : " << sub; cout << "\nPublication : " << pub;
}
};

class officer : public staff
{
protected:
string g;

public:
void oinput()

```

```

{
sinput();
cout << "\nEnter grade : "; cin >> g;
}

void odisplay()
{
idisplay();
cout << "\nGrade : " << g;
}
};

class typist : public staff
{
protected:
double s;
public:
void tpinput()
{
sinput();
cout << "\nEnter speed : "; cin >> s;
}
void tydisplay()
{
idisplay();
cout << "\nSpeed " << s;
}};

class regular : public typist
{
protected:
double sal;

```

```

public:
void input()
{
tpinput();
cout << "\nEnter monthly salary : "; cin >> sal;
}

void display()
{
tydisplay();
cout << "\nSalary : " << sal;
}

};

class causal : public typist
{
protected:
double sal;

public:
void input()
{
tpinput();
cout << "\nEnter daily salary : "; cin >> sal;
}

void display()
{
tydisplay();
cout << "\nSalary : " << sal;
}

};

int main()

```



```

{
int ch, ch2;

cout << "\nEnter 1 for teacher"; cout << "\nEnter 2 for typist"; cout << "\nEnter 3 for officer"; cout <<
"\nEnter your choice : "; cin >> ch;

if (ch == 1)
{
teacher t; t.tinput();
t.tdisplay();
}
else if (ch == 3)
{
officer o; o.oinput();
o.odisplay();
}
else if (ch == 2)
{
cout << "\nEnter 1 for regular"; cout << "\nEnter 2 for causal"; cout << "\nEnter your choice : "; cin >> ch2;
if (ch2 == 1)
{
regular r; r.rinput();
r.rdisplay();
}
else if (ch2 == 2)
{
causal c; c.cinput();
c.cdisplay();
}
}
return 0;
}

```

# Lab Exercise 8(Run Time Polymorphism)

## Question 1:

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.

## (A).

Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area using the concept of dynamic binding.

Remember the two values given as input will be treated as lengths of two sides in the case of rectangles and as base and height in the case of triangles and used as follows:

Area of rectangle =  $x * y$

Area of triangle =  $1/2 * x * y$

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <iostream>
```

```
using namespace std;
```

```
class Shape
```

```
{
```

```
protected:
```

```
double x, y;
```

```
public:
```

```
void get_data()
```

```
{
```

```

cout << "Enter the value for x and y" << endl; cin >> x >> y;

}

virtual void display_area()

{

double area; area = x * y;

cout << "Area : = " << area << " sq.unit" << endl;

}

};

class Triangle : public Shape

{

public:

get_data();

void display_area()

{

double area;

area = 0.5 * x * y; //Area of Triangle

cout << "Area of Triangle = " << area << " sq.unit" << endl;

}

};

class Rectangle : public Shape

{

public:

get_data();

void display_area()

{

double area;

area = x * y; //Area of Rectangle

cout << "Area of Rectangle= " << area << " sq.unit" << endl;

}

};

```

```

int main()
{
Shape *B1 = new Triangle; B1->get_data();
B1->display_area();
Shape *B2 = new Rectangle; B2->get_data();
B2->display_area(); return 0;
}

```

### (B).

Extend the Program-1 to display the area of circle. This requires addition of a new derived class 'circle' that computes the area of a circle. Remember, for a circle we need only one value, its radius, but the get\_data() function in the base class requires two values to be passed. (Hint: Make the second argument of get\_data() function as a default one with zero value.)

```

//*****
//This program is developed by Palash Mishra (201B172)
//*****

#include <iostream>
using namespace std;
class Shape
{
protected:
double x, y;

public:
void get_data()
{
cout << "Enter the value for x and y" << endl; cin >> x >> y;
}
virtual void display_area()
{

```

```

double area; area = x * y;

cout << "Area : = " << area << " sq.unit" << endl;

}

};

class Triangle : public Shape
{
public:
get_data();
void display_area()
{
double area;

area = 0.5 * x * y;//Area of Triangle

cout << "Area of Triangle = " << area << " sq.unit" << endl;

}

};

class Rectangle : public Shape
{
public:
get_data();
void display_area()
{
double area;

area = x * y;//Area of Rectangle

cout << "Area of Rectangle =" << area << " sq.unit" << endl;

}

};

class Cirlce : public Shape
{
public:
get_data();

```

```

void display_area()
{
double area;

cout<<"You can put any value of y because Area is independent on y\n"; y = x;
area = 3.14 * x * y;//Area of Circle
cout << "Area = " << area << " sq.unit" << endl;
}
};

int main()
{
Shape *B1 = new Triangle; B1->get_data();
B1->display_area();

Shape *B2 = new Rectangle; B2->get_data();
B2->display_area(); Shape *B3 = new Circle; B3->get_data();
B3->display_area(); return 0;
}

```

(C).

Run the above program with the following modifications:

- Remove the definition of display\_area() from one of the derived classes.
- In addition to the above change, declare the display\_area() as pure virtual in the base class shape.

Comment on the output in each case.

```

//*****

```

```

//This program is developed by Palash Mishra (201B172)

```

```

//*****

```

```

#include <iostream>

```

```

using namespace std; class Shape

```

```

{

```

```

protected:

```

```

double x, y;

public:
void get_data()
{
cout << "Enter the value for x and y" << endl; cin >> x >> y;
}

virtual void display_area() = 0;
};

class Triangle : public Shape
{
public:
get_data();
void display_area()
{
double area;
area = 0.5 * x * y; //Area of Triangle
cout << "Area of Triangle = " << area << " sq.unit" << endl;
}
};

class Rectangle : public Shape
{
public:
get_data();
void display_area()
{
double area;
area = x * y; //Area of Rectangle
cout << "Area of Rectangle = " << area << " sq.unit" << endl;
}
}

```

```

};

class Cirlce : public Shape
{
public:
    get_data();
    void display_area()
    {
        double area;
        cout<<"You can put any value of y because Area is independent on y\n"; y = x;
        area = 3.14 * x * y; //Area of Circle
        cout << "Area = " << area << " sq.unit" << endl;
    }
};

int main()
{
    Shape *B1 = new Triangle;
    B1->get_data();
    B1->display_area();
    Shape *B2 = new Rectangle;
    B2->get_data();
    B2->display_area();
    Shape *B3 = new Cirlce;
    B3->get_data();
    B3->display_area();
    return 0;
}

```



# Lab Exercise 9(File Handling)

## Question 1:

Write a program that creates a text file “TEXT.txt” on the disk. Write text on this file. Read this file and display the following information on the screen in two columns:

- Number of lines
- Number of words
- Number of characters

Strings should be left-justified and numbers should be right-justified in a suitable field width. Also handle the error by displaying suitable error message.

```
/**
//This program is developed by Palash Mishra (201B172)
//**

#include <iostream>
#include <fstream>
#include <iomanip>
using namespace std;

int main()
{
int line = 0, ch = 0, word = 0, res = 1; char li[80], c, wor[20];
ofstream ofile("TEXT.txt", ios::out | ios::app); while (1)
{
cout << "Enter the line" << endl; gets(li);
ofile << li << endl;
cout << "Do you want to continue (0,1)" << endl; cin >> res;
cin.ignore(); if (res == 1)
continue; else
break;
}
}
```

```

ofile.close();

ifstream file("TEXT.txt", ios::in); while (!file.eof())
{
if (file.eof()) break;
file >> wor; word++;
}

file.close();

ifstream infile("TEXT.txt", ios::in); while (!infile.eof())
{

infile.getline(li, 80); if (infile.eof())
break; line++;
}

infile.close();

ifstream ifile("TEXT.txt", ios::in); while (!ifile.eof())
{
if (ifile.eof()) break;
ifile.get(c); ch++;
}

ifile.close();

cout << "\nNUMBER OF LINES: " << setw(30) << line << endl; cout << "\nNUMBER OF words: " <<
setw(30) << word - 1 << endl; cout << "\nNUMBER OF CHARACTERS: " << setw(25) << ch - 1;

return 0;
}

```

### Question 2:

Two file named 'Source1' and 'Source2' contain sorted list of integers. Write a program that reads the contents of both the files and stores the merged list in sorted form in a new file named 'Target'. Also handle the error by displaying suitable error message

```
//*****
```

```
//This program is developed by Palash Mishra (201B172)
```

```
//*****
```

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
int ARR_1[] = {1, 2, 3, 4, 5}, ARR_2[] = {6, 7, 8, 9, 10}, ARR_3[10];
```

```
ofstream fout1, fout2, fout3;
```

```
ifstream fin1, fin2, fin3;
```

```
fout1.open("Source1");
```

```
for (int i = 0; i < 5; i++)
```

```
{
```

```
fout1.write((char *)&ARR_1[i], (sizeof(ARR_1[i])));
```

```
}
```

```
fout1.close(); fout1.open("Source2");
```

```
for (int i = 0; i < 5; i++)
```

```
{
```

```
fout1.write((char *)&ARR_2[i], (sizeof(ARR_2[i])));
```

```
}
```

```
fout1.close(); fin1.open("Source1"); fout1.open("Target");
```

```
for (int i = 0; i < 5; i++)
```

```
{
```

```
fin1.read((char *)&ARR_1[i], (sizeof(ARR_1[i])));
```

```
fout1.write((char *)&ARR_1[i], (sizeof(ARR_1[i])));
```

```
}
```

```
fin1.close(); fin1.open("Source2");
```

```
for (int i = 0; i < 5; i++)
```

```
{
```

```

fin1.read((char *)&ARR_2[i], (sizeof(ARR_2[i])));
fout1.write((char *)&ARR_2[i], (sizeof(ARR_2[i])));
}
fin1.close(); fout1.close();
fin1.open("Target", ios::in); for (int i = 0; i < 10; i++)
{
fin1.read((char *)&ARR_3[i], (sizeof(ARR_3[i]))); cout << ARR_3[i] << endl;
}
fin1.close(); return 0;
}

```

### Question 3:

In a loop, prompt the user to enter name data consisting of a first name, middle initial, last name, and employee number (type unsigned long). Then, using formatted I/O with the insertion (<<) operator, write these four data items to an ofstream object. Don't forget that strings must be terminated with a space or other whitespace character. When the user indicates that no more name data will be entered, close the ofstream object, open an ifstream object, read and display all the data in the file, and terminate the program.

```

//*****
//This program is developed by Palash Mishra (201B172)
//*****

#include <iostream>
#include <fstream>
using namespace std;
int main()
{
string firstname[3]; string middlename[3]; string lastname[3]; unsigned long empno[3]; char ch = 'y';
int count = 0, temp; ifstream inif; ofstream outf;
outf.open("myfile.dat"); for (int i = 0; i < 3; i++)
{
cout << "first name : "

```

```

<< " ";
cin >> firstname[i];
outf << firstname[i] << '\n'; cout << "middlename : ";
cin >> middlename[i];
outf << middlename[i] << '\n'; cout << "Lastname : ";
cin >> lastname[i];
outf << lastname[i] << '\n'; cout << "Employee no. : "
<< " ";
cin >> empno[i];
outf << empno[i] << '\n'; count++;
cout << "Enter more names (y/n)... "; cin >> ch;
if (ch == 'n')
{
break;
}
}
outf.close(); inif.open("myfile.dat"); temp = count;
for (int i = 0; i < temp; i++)
{
inif >> firstname[i];
cout << "First Name = " << firstname[i] << endl; inif >> middlename[i];
cout << "Middle Name = " << middlename[i] << endl; inif >> lastname[i];
cout << "Last name = " << lastname[i] << endl; inif >> empno[i];
cout << "Employee number = " << empno[i] << endl;
}
inif.close(); return 0;
}

```

#### Question 4:

Write a program to read the file and store the lines into an array. Also handle the error by

displaying suitable error message

```
//*****  
//This program is developed by Palash Mishra (201B172)  
//*****  
  
#include <iostream>  
  
#include <fstream>  
  
#include <string>  
  
using namespace std;  
  
int main()  
{  
string str1, str2[4]; int count = 0, k; char ch, chr = 'y'; ofstream outf; ifstream inif;  
outf.open("filestr.dat"); while (chr != 'n')  
{  
cout << "Sentence : ";  
getline(cin >> ws, str1);  
outf.write((char *)&str1, (sizeof(str1))); count++;  
cout << "Enter more(y/n):"; cin >> chr;  
}  
outf.close(); k = count;  
inif.open("filestr.dat"); for (int i = 0; i < k; i++)  
{  
inif.read((char *)&str1, (sizeof(str1))); str2[i] = str1;  
}  
inif.close();  
for (int i = 0; i < k; i++)  
{  
cout << str2[i] << endl;  
}  
return 0;  
}
```

### Question 5:

Write a program to copy a file in another name. Also handle the error by displaying suitable error message

```
//*****  
//This program is developed by Palash Mishra (201B172)  
//*****  
  
#include <iostream>  
#include <fstream>  
#include <string>  
using namespace std;  
int main()  
{  
string str1, str2[4]; int count = 0, k; char ch, chr = 'y'; ofstream outf; ifstream inif;  
outf.open("filestr.dat"); while (chr != 'n')  
{  
cout << "Sentence : "; getline(cin >> ws, str1);  
outf.write((char *)&str1, (sizeof(str1))); count++;  
cout << "Enter more(y/n):"; cin >> chr;  
}  
outf.close(); k = count;  
inif.open("filestr.dat"); for (int i = 0; i < k; i++)  
{  
inif.read((char *)&str1, (sizeof(str1))); str2[i] = str1;  
}  
inif.close();  
for (int i = 0; i < k; i++)  
{  
cout << str2[i] << endl;  
}  
return 0;
```

```
}
```

Question 6:

Write a program to merge two files and write it in a new file. Also handle the error by displaying suitable error message

```
/**
//*****
//This program is developed by Palash Mishra (201B172)
//*****
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main()
{
string str1, str2[4];
int count = 0, k; char ch, chr = 'y'; ofstream outf; ifstream inif;
outf.open("filestr.dat"); while (chr != 'n')
{
cout << "Sentence : "; getline(cin >> ws, str1);
outf.write((char *)&str1, (sizeof(str1))); count++;
cout << "Enter more(y/n):"; cin >> chr;
}
outf.close(); k = count;
inif.open("filestr.dat"); for (int i = 0; i < k; i++)
{
inif.read((char *)&str1, (sizeof(str1))); str2[i] = str1;
}
inif.close();
for (int i = 0; i < k; i++)
{
cout << str2[i] << endl;
}
```



```
}  
return 0;  
}
```

Question 7:

Write a program to encrypt and decrypt a text file. Also handle the error by displaying suitable error message

```
//*****  
//This program is developed by Palash Mishra (201B172)  
//*****  
  
#include <iostream>  
#include <fstream>  
#include <string>  
using namespace std;  
  
int main()  
{  
string str1, str2[4]; int count = 0, k; char ch, chr = 'y'; ofstream outf; ifstream inif;  
outf.open("filestr.dat"); while (chr != 'n')  
{  
cout << "Sentence : "; getline(cin >> ws, str1);  
outf.write((char *)&str1, (sizeof(str1))); count++;  
cout << "Enter more(y/n):";  
cin >> chr;  
}  
outf.close(); k = count;  
inif.open("filestr.dat"); for (int i = 0; i < k; i++)  
{  
inif.read((char *)&str1, (sizeof(str1))); str2[i] = str1;  
}  
inif.close();
```

```
for (int i = 0; i < k; i++)  
{  
    cout << str2[i] << endl;  
}  
return 0;
```

Palash Mishra(201B172)