
## Data Structures and Algorithms Using C++

**Lab – 1 (2 Hrs Real Time)**

1.1 Write a C + + program to display any message.

1.2 Sum of digit

- 1.2.1 Reverse the number
- 1.2.2 Factorial of a number
- 1.2.3 Fibonacci series
- 1.2.4 Armstrong number checking
- 1.2.5 Prime no checking
- 1.2.6 Palindrome Checking
- 1.2.7 Odd or even number.
- 1.2.8 Perfect number checking.

**Lab – 2 (2 Hrs Real Time)**

2.1 Write a program to calculate the following:
2.2 Find Maximum of N numbers.
2.3 Find Minimum of N numbers.
2.4 Find Summation of N numbers.
2.5 Find Average of N numbers.

**Lab – 3 (2 Hrs Real Time)**

3.1 Write a program to convert the pounds to kilograms using inline function.
3.2 Write a program to using default arguments.
3.3 Write a program to give one example for global or external variables.
3.4 Write a program to give one example for static variables.

**Lab – 4 (2 Hrs Real Time)**

Write a program to do the following:

4.1 String copy
4.2 String concatenation
4.3 String Comparison
4.4 String reverse
4.5 Find the length of the string
4.6 String Conversion (Uppercase to Lowercase & Lowercase to Uppercase)

**Lab – 5 (2 Hrs Real Time)**

Write a program for calculating matrices operations:

5.1 Addition
5.2 Subtraction
5.3 Multiplication
5.4 Transpose of matrices
5.5 Row wise, column wise & diagonal wise total.
5.6 Symmetric Checking.

**Lab – 6 (2 Hrs Real Time)**

6.1 Imagine a tollbooth at a bridge car passing by the booth is expected to apply a fifty-cent tool. Mostly they do, but sometime a car goes by without paying. The tollbooth keeps track of the number of cars that have gone by and of the total amount of money collected.

Model this tollbooth with a class called tollbooth. The two data items are a type-unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both these to 0. A member function called payingcar increments the car total and adds 0.50 to the cash total. Another function, but nopaycar( ), increments the car total but adds nothing to the cash total. Finally, a member function called display ( ) displays the two totals.

Include a program to test this class. This program should allow the user to push one key to count a paying car and another to count a non-paying car. Pushing the ESC key should cause the program to print out the total cars and total cash and then exit.

6.2 Write a Program to create a class called student in this class we have one member function. It will used to display the attributes of student class. The attributes of student class are student rollno, name, result, and percentage.

6.3 Write a program define a class to represent a bank account include the following data members name of the depositor, account no, balance amount

1. Assign initial value
2. Deposit the amount
3. Withdraw amount after checks the balance
4. Display the name & balance write a program with test 10 customers.

6.4 Write a C+ + program to read one text and find the following options.

count number of characters
count number of words
count number of lines
searching a character
search a substring
replace a string
deleting the word
Insertion of a word.

6.5 Write a C+ + program to create a person class and find the total, average and grade of each student and count the grade of I, II, & III, display the report neatly.

6.6 Write a program to create a class called staff. The attribute of the class is name of the staff, data of appointment and qualification. In this class were have two member functions one is used for get the input and another one is used for display the output.

6.7 Write a program to define a class called room with the following attributes 1.length 2.breadth, 3.height, 4.floor area, 5. Wall area, 6. No of fans. 7. No of windows 8. Number of doors. Define a suitable constructor and a method to display details of a room.

**Lab – 7 (2 Hrs Real Time)**

7.1 Create a class called Time that has separate int member data for hours, minutes and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. A member function should display it, in 11:59:59 format. The final member function should add two objects of type Time passed as arguments.

A main ( ) program should create two initialized Time objects, and one that isn't initialized. Then it should add the two initialized values together, leaving the result in the third Time variable finally it should display the value of this third variable.

7.2 Write a program to create a class complex in this class do the calculation of complex numbers.

1. Addition
2. Subtraction
3. Multiplication
4. Division (using constructor and friend function )

7.3 Create two classes DM & DB which store the value of distances DB stores distance in meters and cenTimeters and DB in feet and inches. Write a program that can read values for the class objects and add one objects DM with another objects of DB use a friend function to carry out the addition operation. The object that stores the result may be an DM objects or DB object depending on the s in which the results are required .

The display should be in the format of feet and inches or meters in cenTimeters depending on the objects on display.

7.4 Write a program to using the user-defined manipulations.

7.5 Write a program to using the manipulator display any output neatly.

**Lab – 8 (2 Hrs Real Time)**

8.1 Give a suitable example for Single inheritance.
8.2 Give a suitable example for multiple Inheritance.

8.3 Give a suitable example for multi level Inheritance.

8.4 Give a suitable example for Hybrid Inheritance.

8.5 Give a suitable example for Hierarchical Inheritance.

8.6 Design a class "university". It has name and place as data members. Their member functions are for printing and getting data members. Derive 'Department' class from university class protected. Its data members are department name, no of students, and number of staffs. Member functions are for getting and printing data members.

Derive one more class staff from department data members are name of staff, salary, experience (in years) and subject. Another class student is derive from department as student. Data members are name of student, years of course, course and total marks. These two classes are derived from parent class in private mode. Write a C+ + program for the above situation.

8.7 Define the base class person holding information as name, date of birth, sex, height & weight. Derive two classes from this base class in public mode as major and minor. In major class data members are as status (married / unmarried /widow) and job. In Minor class data members are as father name. Derive another class, which contains information as hobby. Declare base class as virtual base class. Write a program for above situation and process these information.

8.8 Define a class patient with data members as name of patient sex and age. Derive a class description with members ward no, bed no, nature of illness and date of admission another class 'doctor' is derived from base class patient has information as name of doctor and qualification of doctor. The class is derived from these two classes having information condition [normal/critical] note: using constructor for derived classes.

8.9 Define a class player data members are name and nationality. Derive the class event. It is data members event type and [individual/ team] name of event. Define another derived class from player named personnel, data members are other personal information. Write a function printing for all three classes base class function is virtual function. By getting the choice we can either display event details or personal details.

8.10 Define a base class contains information as book-code, book-title and author of book. Second base class contains the data members as publication year and publisher name. The derived class has the members as price of book and availability member functions are as follow. Entering a new book, editing the given book, listing all the books, list only the available books sorting the books and exit.

**Lab - 9 (2 Hrs Real Time)**

9.1 Write a program using unary operator create a counter.

9.2 Write a program to create a class shape. In this class find the area of :

1. Circle
2. Square
3. Rectangle
4. Triangle (using function over loading concepts)

9.3 Define a class 'date' using operator overloading [<] check the given dates. If the first date is greater than second date then subtract the second date from first date else subtract the first date from the second date using operator [-=].

9.4 Using the operator overloading perform the operations of stack and queue, operation using [()].

**stack** : pop = stack();
push = stack(value);
**queue** : insert = queue(value);
delete = queue();
Using function overloading also.

9.5 Write a program to calculate the operations of string using operator over loading.

9.6 Create a base class called 'shape' use this class to store two double type values that could be used to compute area of figures. Derive two specific classes called triangle and rectangle from the base class shape. Add to a

base class, a member function getdata() to initialize base class data members and another member function display_area() to compute and displays area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirement.

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

Derive another class circle for computing area of circle. getdate() function has two arguments. Since circle needs only one argument, second argument of getdata ( ) is a default one with zero value Note: using virtual function concept.

9.7 Write a program to create a virtual function demonstration using run Time binding.

9.8 Write a program for static binding (accessing member function using pointers)

9.9 Write a program for dynamic binding (accessing member function using pointers)

9.10 Write a program for pure virtual function.


## MS1.2 Data Structues and Algorithms Using C++

**Lab Unit – 1 (2 hrs Real Time)**

# Ex - 1.1

```
include <iostream.h>
void main()
{
    cout << " Welcome to C++ and Java " << endl;
    }
```

# Ex - 1.2.1

```
      /* sum of digit */

#include <iostream.h>
#include <conio.h>
void main()
{
       int s,n,n1;
       cout <<"enter the number \n";
       s=0;
       cin >>n;
       cout <<"the given number is  \n"<<n;
       a: do
        { n1=n%10;
          s=s+n1;
          n=n/10;
          }while (n!=0);
          if(s>9)
          { n=s;
            s=0;
            goto a;
          }
          else
          cout <<"sum of digit is \n"<<s;
          getch();
       }
```

# Ex - 1.2.2

```
/* reverse the number */

#include <iostream.h>
```

```
#include <conio.h>
void main()
{
       int r, n;
       cout <<"enter the number \n";
       cin >>n;
       cout <<"reversed number is ";
        do
        {
         r=n%10;
         n=n/10;
         cout <<r;
         }while(n!=0);
         cout << "\n";
         getch();
         }
```

## Ex 1.2.3

```
             /*  factorial of a number */
#include <iostream.h>
#include <conio.h>
void main()
{
       int i,f,n;
       f=1;
       cout <<"enter the number \n";
       cin >>n;
       for(i=1;i<=n;i++)
       f=f*i;
       cout <<"Factorial of number is "<<f;
       getch();
  }
```

## Ex. - 1.2.4

```
                /* Fibonacci series */
#include <iostream.h>
#include <conio.h>
void main()
{
       int l, i, j, k, s;
     cout <<"enter the total number \n";
     cin >>k;
     cout <<"the total number of the series is  \n"<< k;
     i=0,j=1;
     cout <<i<<j;
     for(l=3;l<=k;l++)
     {s=i+j;
      cout <<"\n"<<s;
      i=j;j=s;
      }
      getch();
    }
```

# Ex - 1.2.5

```
                /* Armstrong number */
#include <iostream.h>
#include <conio.h>
void main()
{
       int s, n, n1, r;
     cout <<"enter the number \n";
     cin >>n;
     cout <<" the given number is \n"<< n;
     s=0;
     n1=n;
     do
     {
      r=n%10;
      n=n/10;
      s=s+r*r*r;
      }while (n!=0);
      if(s==n1)
       cout <<"given no is armstrong\n";
      else
       cout <<"given no is not armstrong\n";
      getch();
       }
```

**Ex 1.2.6:**

```
                      /* prime number */
#include <iostream.h>
#include <conio.h>
void main()
{
       int l, i, j, k;
       clrscr();
       l=0;
       cin >>n;
       for(i=1;i<=n;i++)
        {
        for(j=2;j<=(i-1);j++)
         {
          k=i%j;
          if (k==0)
           l=1;
          else
           l=l;
           }
        if (l==0)
          cout <<"prime \n"<<i;
      else
        cout <<"not prime \n"<<i;
      l=0;
       }
       getch();
       }
```

**Ex 1.2.7:**

```
                      /* palindrome  or not */
```

```
#include <iostream.h>
#include <conio.h>
void main()
{
       int n,n1,s,r;
     cout <<"enter the number\n";
     cin >>n;
     cout << "the given number is  \n"<<n;
     n1=n;
     s=0;
     do
     {
      r=n%10;
      n=n/10;
      s=s*10+r;
      } while(n!=0);
       if(s==n1)
        cout <<"the given number is palindrome  \n"<<n1;
       else
    cout <<"the given number is not palindrome  \n"<<n1;
    getch();
    }
```

**Ex 1.2.8**

```
                /* odd or even number */
#include <iostream.h>
#include <conio.h>
void main()
{
       int n,n1;
       cout <<"enter the number \n";
       cin >>n;
       cout <<"the given number is  \n"<<n;
       n1=n%2;
       if(n1==0)
          cout << n <<"  is the even number \n";
      else
        cout <<n<<"  is the odd number \n";
      getch();
}
```

# Ex 1.2.9

```
             /*  Perfect number Checking */
#include <iostream.h>
#include <conio.h>
void main()
{
       int i,f,n;
       cout <<"enter the number \n";
       cin >>n;
       for(i=1;i<n;i++)
       {
```

```
            if (n%i == 0 )
            f=f+i;
           }
           if (f == n)
             cout <<" Number is Perfect ";
           else
             cout <<" Number is not perfect ";
           getch();
  }
```

**Lab Unit – 2 (2 hrs Real Time)**

**Ex - 2.1**

```
                    /* Biggest among 'n' number */

#include <iostream.h>
#include <conio.h>
void main()
{
      cout <<"enter the total number \n";
      cin >>n;
      cout <<"total number of array is \n"<<n;
      for(i=1;i<=n;i++)
      cin >>a[i];
      cout <<"the array is \n";
      for(i=1;i<=n;i++)
      cout <<"\n"<<a[i];
      g = a[1];
      for(i=2;i<=n;i++)
      {
        if(g<a[i])
       g = a[i];
       }
     cout <<"greatest among 'n' number is \n"<< g;
     getch();
  }
```

# Ex - 2.2

```
          /* Smallest among 'n' number */
#include <iostream.h>
#include <conio.h>
void main()
{
  cout <<"enter the total number\n";
  cin >>n;
  cout <<"total number of array is \n"<<n;
  for(i=1;i<=n;i++)
   cin >>a[i];
  cout <<"the array is \n";
  for(i=1;i<=n;i++)
   cout <<"\n"<<a[i];
```

```
  s = a[1];
  for(i=2;i<=n;i++)
  {
   if(s>a[i])
     s = a[i];
     }
     cout <<"smallest among 'n' number is  \n"<< s;
     getch();
  }
```

## Ex - 2.3

```
                  /* Summation of 'n ' number */
#include <iostream.h>
#include <conio.h>
void main()
{
  cout <<"enter the total number \n";
  cin >>n;
  cout <<"total number of array is  \n"<<n;
  for(i=1;i<=n;i++)
    cin >>a[i];
  s=0;
  cout <<"the array is \n";
  for(i=1;i<=n;i++)
   cout <<"\n"<<a[i];
  for(i=1;i<=n;i++)
   s=s+a[i];
 cout <<"Summation of 'n' number is  \n"<< s;
     getch();
}
```

## Ex - 2.4

```
           /* Average of 'n ' number */

#include <iostream.h>
#include <conio.h>
void main()
{
  cout <<"enter the total number \n";
  cin >>n;
  cout <<"total number of array is  \n"<<n;
  for(i=1;i<=n;i++)
   cin >>a[i];
  s=0;
  cout <<"the array is \n";
  for(i=1;i<=n;i++)
   cout <<"\n"<<a[i];
  for(i=1;i<=n;i++)
   s = s + a[i];
   s = s / n;
  cout <<"average of 'n' number is  \n"<< s;
     getch();
}
```

**Lab Unit – 3 (2 hrs Real Time)**

**Ex – 3.1**

```
# include <iostream.h>
 inline float ptokg (float po)
{
return 0.453 * po;
   }
void main  ( )
   {
       float p;
       cout << enter four weight in pounds;
       cin  >> p;
       cout << "your weight in kilograms in" << ptokg (p);

      }
```

## Ex – 3.2

```
# include <iostream.h>
void repeat (Char  = "*", int = 45);
void main ( )
{
   repeat ( );                 / / 45 asterisks
   repeat ("=");             / / 45 equals
   repeat ("+",30);        / / prints 30 plus
}

void repeat (char ch, int n)
 {
    for ( int  i=0; i<n; i+ +)
    cout << ch;
     cout << endl;
    }
```

## Ex – 3.3

```
Global variables
# include <iostream.h>
# include <conio.h>
char ch = 'a';
void  get();
void put();
void main()
     {
        while (ch ! = '\r')
         { get();
            put();
            }
    }
```

```
void get()
  {
     ch = getch();
        }
  void put()
  {
    cout <<ch;
     }
```

# Ex - 3.4

```
# include <iostream.h>
  float get (float);
  void main ()
   {
     float d = 1,a;
      While (d != 0)
         {  cout << "Enter a number";
            cin  >> d;
            a = get(d);
            cout << "new average in " << a << endl;
            }
    }
float get(float n)
     {
        static float t =0;
        static int  c = 0;
        c++;
        t += n;
        return t/c;
        }
```

**Lab Unit – 4 (2 hrs Real Time)**

**Ex - 4.1**

```
// strcpy.cpp: copying string
# include < iostream.h>
# include < string.h>
void main ( )
{
   char s1[25], s2 [25];
     cout << "Enter a string:" ;
    cin >> s1;
    strcpy( s2, s1);
     cout << "strcpy (s2, s1) : "< < s2;
}
```

# Ex - 4.2

```
 / / strcat.cpp: string concatenation
 # include < iostream.h>
 # include < string .h>
 void main ( )
```

```
 {
       char s1[40], s2 [25];
       cout < < “Enter string s1;”;
       cin >> s1;
       cout < < “Enter string s2:”;
       cin >> s2;
       strcat(s1,s2);
       cout < < “strcat (s1, s2 ) : “< < s1;
}
```

## Ex - 4.3

```
/ / strcmp.cpp: string comparison
# include < iostream.h>
# include < string.h>
void main ( )
{
     char s1[25], s2 [25];
     cout << “Enter string s1:”;
     cin >> s1;
     cout << “Enter string s2:”;
     cin >> s2;
     int status  = strcmp ( s1 ,s2 );
     cout < < “strcmp (s1,  s2 ):”;
     if (status  = = 0)
       cout < < “s1 < <” is greater than” << s2;
     else
       cout < < s1 < < “is less than” << s2;
}
```

## Ex - 4.4

```
// strrev.cpp: Reverse string
# include < iostream.h>
# include < string.h>
void main ( )
{
   char s1[25], s2 [25];
   cout << “Enter a string:” ;
   cin >> s1;
   s2 = strrev ( s1);
   cout << “strrev (s1) : “< < s2;
}
```

## Ex - 4.5

```
/ / Strlen.cpp: computing length of string
# include <iostream.h>
# include <string.h>
void main ( )
{
     char s1[25];
     cout << “Enter your name:”;
```

```
    cin > > s1;
    cout < < “Strlen(s1): “< < strlen(s1) < < end1;
  }
```

# Ex - 4.6

```
/ / uprlwr.cpp: converting string to upper or lower case
# include < iostream.h>
# include < string.h>
void main ( )
{
     char s1 [25], temp[25];
     cout < < “Enter a string:”;
     cin > > s1;
     strcpy ( temp, s1);
     cout < < “strupr ( temp ) : “ < < strupr (temp) < < end1;
     cout < < “strlwr (temp)   : “ < < strlwr ( temp) < < end1;
}
```

**Lab Unit – 5 (2 hrs Real Time)**

**Ex - 5.1**

```
      /* Matrix addition */

#include <stdio.h>
#include <conio.h>
void main()
 {
   int a[10][10],b[10][10],c[10][10],n, m, i, j;
   printf("enter the row and column \n");
   printf("-------------------------\n");
   scanf("%d %d", &n, &m);
   printf("enter the first matrix \n");
   printf("-----------------------\n");
   for(i=1;i<=n;i++)
   {
    for(j=1;j<=m;j++)
     scanf("%d", &a[i][j]);
     }
  printf("enter the second matrix \n");
  printf("-------------------------\n");
  for(i=1;i<=n;i++)
   {
    for(j=1;j<=m;j++)
     scanf("%d", &b[i][j]);
    }
   for(i=1;i<=n;i++)
   { for(j=1;j<=m;j++)
      c[i][j]=a[i][j]+b[i][j];
    }
    printf("print the resultant matrix \n");
    printf("---------------------------\n");
    for(i=1;i<=n;i++)
     {for(j=1;j<=m;j++)
      printf("%d \t", c[i][j]);
```

```
     printf("\n");
    }
    getch();
    }
```

## Ex - 5.2

```
          /* matrix subtraction */
#include <stdio.h>
#include <conio.h>
void main()
 {
 int a[10][10],b[10][10],c[10][10],n, m, i, j;
 printf("enter the row and column \n");
 printf("-------------------------\n");
 scanf("%d %d", &n, &m);
 printf("enter the first matrix \n");
 printf("-------------------------\n");
 for(i=1;i<=n;i++)
  { for(j=1;j<=m;j++)
     scanf("%d", &a[i][j]);
   }
  printf("enter the second matrix \n");
  printf("-------------------------\n");
  for(i=1;i<=n;i++)
  {for(j=1;j<=m;j++)
   scanf("%d", &b[i][j]);
   }
   for(i=1;i<=n;i++)
    {for(j=1;j<=m;j++)
     c[i][j]=a[i][j]-b[i][j];
    }
    printf("print the resultant matrix \n");
    printf("---------------------------\n");
    for(i=1;i<=n;i++)
    {for(j=1;j<=m;j++)
     printf("%d \t", c[i][j]);
     printf("\n");
     }
     getch();
     }
```

## Ex - 5.3

```
         /* matrix multiplication */
#include <stdio.h>
#include <conio.h>
void main()
 {
 int a[10][10],b[10][10],c[10][10],n, m, o, p, i, j, k;
 printf("Enter the row and column for first matrix \n");
 printf("-----------------------------------------\n");
 scanf("%d %d", &n, &m);
 printf("Enter the row and column for second matrix \n");
 printf("-----------------------------------------\n");
 scanf("%d %d", &o, &p);
```

```
if(m == o)
{
 printf("Enter the first matrix \n");
 printf("-----------------------\n");
for(i=1;i<=n;i++)
{for(j=1;j<=m;j++)
 scanf("%d", &a[i][j]);
 }
 printf("Enter the second matrix \n");
 printf("------------------------\n");
 for(k=1;k<=o;k++)
 {for(l=1;l<=p;l++)
  scanf("%d", &b[k][l]);
  }
  for(i=1;i<=n;i++)
  {for(j=1;j<=p;j++)
   {for(k=1;k<=o;k++)
    c[i][j]=c[i][j]+a[i][k]*b[k][j];
   }
   }
   printf("The resultant matrix \n");
   printf("--------------------\n");
   for(i=1;i<=n;i++)
   {for(j=1;j<=p;j++)
    printf("%d \t", c[i][j]);
    printf("\n");
    }
    }
    else
printf("Multiplication is not possible for the given data \n");
printf("-------------------------------------------------\n");
   getch();
    }
```

## Ex - 5.4

```
         /* matrix transpose */
#include <stdio.h>
#include <conio.h>
void main()
 {
 int a[10][10],b[10][10],n, m, i, j;
 printf("enter the row and column \n");
 printf("-------------------------\n");
 scanf("%d %d", &n, &m);
 printf("enter the matrix \n");
 printf("-----------------\n");
  for(i=1;i<=n;i++)
 {for(j=1;j<=m;j++)
  scanf("%d", &a[i][j]);
  }
  printf("the given matrix is \n");
  printf("--------------------\n");
  for(i=1;i<=n;i++)
  {for(j=1;j<=m;j++)
  printf("%d \t", a[i][j]);
  printf("\n");
  }
```

```
  for(i=1;i<=n;i++)
  {for(j=1;j<=m;j++)
   b[j][i]=a[i][j];
   }
   printf("print the resultant matrix \n");
   printf("--------------------------\n");
   for(i=1;i<=m;i++)
   {for(j=1;j<=n;j++)
    printf("%d \t", b[i][j]);
    printf("\n");
    }
    getch();
    }
```

**Ex - 5.5**

```
      /* Row wise, column wise and diagonal wise total */
#include <stdio.h>
#include <conio.h>
void main()
 {
 int a[10][10],n,m,i,j,r1,s1,d;
 printf("enter the row and column \n");
 printf("-------------------------\n");
 scanf("%d %d", &n, &m);
 printf("enter the matrix \n");
 printf("-------------------\n");
 for(i=1;i<=n;i++)
 {for(j=1;j<=m;j++)
  scanf("%d", &a[i][j]);
  }
  printf("given matrix is \n");
  printf("----------------\n");
 for(i=1;i<=n;i++)
 {for(j=1;j<=m;j++)
  printf("%d \t", a[i][j]);
  printf("\n");
  }
  r1=0;
  s1=0;
  printf("The Rowwise and columnwise Summation \n");
  printf("------------------------------------ \n");
  printf("\n");
  for(i=1;i<=n;i++)
 {
  for(j=1;j<=m;j++)
   {
     r1=r1+a[i][j];
     printf("%d \t", a[i][j]);
   }
   printf("| %d",r1);
   printf("\n");
   r1=0;
  }
  printf("|\n");
  printf("-------------------\n");
  for(i=1;i<=n;i++)
 {
```

```
 for(j=1;j<=m;j++)
 {
 s1=s1+a[j][i];
  }
  printf("%d\t",s1);
  s1=0;
}
d=0;
for(i=1;i<=n;i++)
 {
  d = d + a[i][i];
 }
printf("\n \n");
printf("diagonalwise summation  is  %d \n", d);
printf("-----------------------------\n");
getch();
}
```

# Ex - 5.6

```
/*  symmetric checking */

#include <stdio.h>
#include <conio.h>
void main()
 {
 int a[10][10],n, y, i, j;
   printf("enter the order of the matrix \n");
   printf("-------------------------------\n");
   scanf("%d %d", &n, &y);
   printf(" the order of the matrix is %d %d \n", n, m);
   printf("---------------------------------------\n");
   printf(" enter the value of the matrix \n");
   printf("-------------------------------\n");
   for(i=1;i<=n;i++)
   {
    for(j=1;j<=y;j++)
    scanf("%d", &a[i][j]);
    }
    printf("the given matrix is \n");
    printf("---------------------\n");
   for(i=1;i<=n;i++)
   {
    for(j=1;j<=y;j++)
    printf("%d \t", a[i][j]);
    printf("\n");
    }
    printf("the transpose of matrix is \n");
    printf("--------------------------\n");
   for(i=1;i<=n;i++)
   {
    for(j=1;j<=y;j++)
    printf("%d \t", a[j][i]);
    printf("\n");
    }
    for(i=1;i<=n;i++)
```

```
    {
    for(j=1;j<=y;j++)
    {
     if (I != j)
     if (a[i][j] != a[j][i])
     goto d;
     }
     }
      printf("the given matrix is symmetric \n");
      printf("-----------------------------\n");
      goto f;
   d: printf("the given matrix is not symmetric \n");
      printf("---------------------------------\n");
   f: getch();

 }
```

**Lab Unit – 6 (2 hrs Real Time)**

**Ex – 6.1**

```
# include <iostream.h>
# include <conio.h>
const char Esc  = 27;
const double Toll = 0.5;
class tollbooth
    {  private :
        unsigned int totalcars;
        double tc;
        public:
               tollbooth()
                {
                   totalcars = 0;
                   tc = 0.0;
                 }
void payingcar()
   {
      totalcars + +;  +c  + = Toll; }
      void nopaycar ( )
       {
         totalcars + + ;
         }
void display()
     {
      cout << " \n cars =" totalcars <<" cash =" << tc ;
      }
       };
void main()
 {     tollbooth b1;
       char ch;
       cout << "\n press 0 for each non –paying car"
                << "\n   1 for each paying car"
                << "\n  Esc to exit the program";
       do
       {
          ch = getch();
           if (ch == '0')
              b1.nopaycar();
```

```
                if (ch == '1')
                  b1.payingcar();
            } while (ch != Esc);
          b1.display();
             }
```

## Ex - 6.2

```
        / * Student class */

#include <iostream.h>
#include <conio.h>
void main()
 {
class student
{
private :
int roll;
float perc;
char name[25],result[6];
public:
void display();
};
void student :: display()
{
cout << " Enter the details of student \n";
cout << " Enter the Rollno, Name, Result and Percentage \n";
cin >> roll >> name >> result >> perc;
cout << " Student Details \n";
cout << " Roll number " << roll << endl;
cout << " Name  " << name << endl;
cout << " Result " << result << endl;
cout << " Percentage " << perc << endl;
}
void main()
{
student s;
s.display();
}
```

## Ex 6.3

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
class bank
       {
        private:
               char name[20];
               long acbal;
        public:
               int with, amt, i, n, acnum, acno;
               void input();
               void deposit();
               void withdraw();
               void display();
       };
```

```
void bank::input()
{
  cout << " CUSTOMER DETAILS \n";
  cout << " ================ \n";
  cout << " Enter customer Name : ";
  cin >> name;
  cout << " Enter A/C No : ";
  cin >> acno;
  cout << " Enter A/C Balance : ";
  cin >> acbal;
  }

void bank::deposit()
{
      if(acbal < 0)
        cout<<"ERROR \n";
      else
       {
         cout<<"Enter the Amount to be deposited : ";
         cin>>amt;
         acbal = acbal + amt;
         cout << " Customer Details \n";
         cout << "==================\n";
         cout << "NAME : " << name << "\n" << "A/C NO : " << acno;
         cout << "\n" << "A/C BAL : " << acbal;
         cout << "\n=================\n";
       }
}
void bank::withdraw()
{
      if (acbal < 100)
        cout<<"ERROR \n";
      else
       {
         cout << "Enter the amount to be withdrawn : ";
         cin >> with;
         if(acbal<with)
            cout << "\n AMOUNT GREATER THAN BALANCE \n";
         else
         {
         acbal = acbal - with;
         cout << " Customer Details \n";
         cout << "==================\n";
         cout << " NAME : "<< name <<"\n" << " A/C NO : " << acno;
         cout << "\n" << " A/C BAL : " << acbal;
         cout << "\n==============\n";
       }
     }
}
void bank::display()
{
       cout << " Customer Details \n";
       cout << " ================ \n";
       cout << " NAME : "<< name << "\n" << " A/C NO : " << acno;
       cout << "\n"<<" A/C BAL : " << acbal;
       cout << "\n=================\n";
```

```
    }
void main()
{
 bank b[10];
 int i, n, acnum, ch, op;
 clrscr();
 cout << " Enter the no of Customers : ";
 cin >> n;

 for(i=0;i<n;i++)
 {
  b[i].input();
 }
as: cout << "\n Enter the A/C no in which U want to do transactions: ";
  cin >> acnum;
  for(i=0;i<n;i++)
  {
  if(b[i].acno == acnum)
  {
 op=1;
 while (op==1)
{
 cout << "\t\t\t MENU \n";
 cout << "\t\t |==============|\n";
 cout << "\t\t | 1 : DEPOSIT  |\n";
 cout << "\t\t | 2 : WITHDRAW |\n";
 cout << "\t\t | 3 : DISPLAY  |\n";
 cout << "\t\t |==============|\n";
 cout << "\n\n Please, Enter U'r choice : \t";
 cin >> ch;
 cout << "\n";
 switch(ch)
 {
  case 1:
        cout << " DEPOSIT \n";
        cout << "=========\n";
        b[i].deposit();
        break;
  case 2:
        cout << " WITHDRAW \n";
        cout << "==========\n";
        b[i].withdraw();
        break;
  case 3:
        cout << " DISPLAY \n";
        cout << "=========\n";
        b[i].display();
        break;
 }
 cout<<"\n To continue any above process, use [1] : ";
 cin >> op;
 }
 }
 }
 cout << "\n Do U, want to do any changes in other records, use [1] \t";
 int co;
 cin >> co;
```

```
 cout << "\n";
 if (co == 1)
 goto as;
}
```

## Ex – 6.4

```
// program to count number of words, characters, lines in a text

#include<iostream.h>
#include<string.h>
#include<conio.h>
#include<ctype.h>
#include<process.h>
#include<stdio.h>
class count
  {  private:
     char *a, g;
     int l;
     public:
       count()
        { l = 0;
          a = new char[l + 1];
        }
       ~count()
        {
           delete a;
        }
       void getdata();
       void lines();
       void words();
       void display();
       void character();
       void printdata();
       void searchar();
       int cpattern(char s[]);
       void replace();
       void remove();
       void insert();
    };

void count::getdata()
  {  int i= 0;
     a[0] = '\0';
     cout<<" \n\t\t Reading the text \n";
     cout<<"\n\n Enter the text . Press @ to stop inputting \n";
     a[0] = '\0';
     while( (g = getchar()) != '@' )
       {  a[i++] = g; }
     a[i] = '\0';
     l = i;
   }
void count::character()
  {  int nc = 0;
     for(int i = 0; i < l; i++)
       if ( !isspace(a[i]) )
```

```
                nc++;
      cout<<" \n\n\n\t\t Number of Characters =  "<<nc;
      cout<<"\n\n Press any key to continue \n";
      getch();
     }
void count::lines()
  {  int nl = 1;
     for(int i = 0; i < l; i++)
if ((a[i] == '\n') && (a[i+1] != '\0' && (a[i] == '\n' && a[i+1]
                    != '\n'  && i != 0 )) )
          nl++;
     cout<<" \n\n\n\t\t Number of Lines  =  "<<nl;
     cout<<"\n\n Press any key to continue \n";
     getch();
  }

void count::words()
 {   int nw = 1;
     for(int i = 0; i <= l ; i++)
     if ( a[i] == '' )
{if ( a[i + 1 ] != '' && a[i + 1] !=  '\0' && a[i+ 1] != '\n')
       nw++;}
     else
       if ( a[i] == '\n' && isalnum(a[i + 1])  && i != 0 )
            nw++;
     cout<<" \n\n\n\t\t Number of Words =   "<<nw;
     cout<<"\n\n Press any key to continue \n";
     getch();
    }

void count::display()
  {   cout<<"\n\n\n\t\t";
      for(int i = 0; i < l; i++)
        { cout<<a[i];
          if ( a[i] == '\n' )
               cout<<"\t\t";
        }
     cout<<"\n\n Press any key to continue \n";
     getch();
   }

void count::searchar()
 { char s;
   cout<<"\n\t\t Searching the character \n";
   cout<<"\t\t ------------------------";
   cout<<"\n\n Enter the character to be searched \n";
   cin>>s;
   clrscr();
   cout<<"\n\n\n\t\t Character to be searched is  :  "<<s;
   int c = 0;
   for(int i = 0; i < l; i++)
     if ( a[i] == s )
            c++;
   if ( c == 0 )
      cout<<" \n\t\t Character "<<s<<" Not found ";
   else
   cout<<"\n\n\t\t Number of  Occurrence :"<<c;
```

```
   cout<<"\n\n\n Press any key to continue \n";
   getch();
  }

int count::cpattern(char s[])
 { int c = 0,j = 0;
   char r[15];
   for(int i = 0;i < l + 1; i++)
    {
      if ( a[i] != ' ' && a[i] != '\n' && a[i] != '\0')
           {  r[j] = a[i];
              j++;
           }
      else
         { r[j] = '\0';
           j = 0;
           if ( strcmp(r, s) == 0 )
              c++;
         }
     }
   return(c);
  }

void count::replace()
  { int c = 0,j = 0,p1;
    char s[15],r[15],p[15],t[200],temp[2];
    cout<<"\n\t\t  Replacing the pattern ";
    cout<<"\n\t\t ----------------------";
    cout<<"\nEnter the Pattern to be Replaced \n";
    cin>>s;
    p1 = cpattern(s);
    if ( p1 == 0 )
       cout<<"\n\n pattern not found \n";
    else
    {
    cout<<" Enter the Pattern for replacement \n";
    cin>>p;
    clrscr();
    t[0] = '\0';
    for(int i = 0;i < l; i++)
    {

      if ( a[i] != ' ' && a[i] != '\n' && a[i] != '\0')
           {  r[j] = a[i];
              j++;
           }
      else
         {
           r[j] = '\0';
           j = 0;
           if ( strcmp(r, s) != 0 )
               strcat(t, r);
           else
              strcat(t, p);
           temp[0] = a[i];
           temp[1] = '\0';
           strcat(t, temp);
```

```
            r[0] = '\0';
           }
      }
      strcpy(a, t);
      l = strlen(a);
      cout<<"\n\n\t\tString after  replacement \n";
      cout<<" \t\t------------------------ ";
      display();
      }
 }

void count::remove()
  { int c = 0,j = 0,p1;
    char s[15],r[15],t[200],temp[2];
    cout<<"\n\t\t  Removing the pattern ";
    cout<<"\n\t\t ----------------------";
    cout<<"\nEnter the Pattern to be Removed \n";
    cin>>s;
    p1 = cpattern(s);
    if ( p1 == 0 )
       cout<<"\n\n  Pattern not found \n";
    else
    {
    clrscr();
    t[0] = '\0';
    for(int i = 0;i < l; i++)
    {
      if ( a[i] != ' ' && a[i] != '\n' && a[i] != '\0')
           {  r[j] = a[i];
              j++;
            }
      else
         {
           r[j] = '\0';
           j = 0;
           if ( strcmp(r, s) != 0 )
               strcat(t, r);
           temp[0] = a[i];
           temp[1] = '\0';
           strcat(t, temp);
           r[0] = '\0';
          }
     }
     strcpy(a, t);
     l = strlen(a);
     cout<<"\n\n\t\t String after  removing the pattern \n";
     cout<<"\t\t-----------------------------------";
     display();
     }
     cout<<"\n\n Press any key to continue \n";
     getch();

 }

void count::insert()
  { int c = 0,j = 0,p1;
    char s[15],r[15],p[15],t[200],temp[2];
```

```
    cout<<"\n Inserting a pattern";
    cout<<"\n -------------------";
    cout<<"\n Enter the Pattern to be searched \n";
    cin>>s;
    p1 = cpattern(s);
    if ( p1 == 0 )
       cout<<"\n\n  Pattern not found \n";
    else
    {
     cout<<"\n Enter the Pattern to be inserted \n";
     cin>>p;
     t[0] = '\0';
     for(int i = 0;i < l; i++)
     {
      if ( a[i] != ' ' && a[i] != '\n' && a[i] != '\0')
           {  r[j] = a[i];
              j++;
            }
      else
         {
           r[j] = '\0';
           j = 0;
           if ( strcmp(r, s) != 0 )
               strcat(t, r);
           else
              { strcat(t, p);
               strcat(t," ");
               strcat(t, r);
              }
           temp[0] = a[i];
           temp[1] = '\0';
           strcat(t, temp);
           r[0] = '\0';
         }
     }
     strcpy(a, t);
     l = strlen(a);
     cout<<"\n\n\t\t String after  Insertion \n";
     cout<<"\t\t ----------------------- ";
     display();
     }
     cout<<"\n\n Press any key to continue \n";
     getch();
 }

void main()
 { count c;
   clrscr();
   char s[200];
   int opt, temp;
   c.getdata();
   do
   {
    clrscr();
    cout<<"\n\t\t\t  MENU  \n ";
    cout<<" \n\t\t 1.Counting the number of lines ";
```

```
cout<<" \n\t\t 2.Counting the number of words ";
cout<<" \n\t\t 3.Counting the number of characters";
cout<<" \n\t\t 4.Searching a character ";
cout<<" \n\t\t 5.Searching a Substring ";
cout<<" \n\t\t 6.Replacing all occurrences of Word";
cout<<" \n\t\t 7.Deleting all occurrences of Word";
cout<<" \n\t\t 8.Insertion of a Word";
cout<<" \n\t\t 9.Display";
cout<<" \n\t\t 10.Exit";
cout<<" \n\n Enter your Choice";
cin>>opt;
clrscr();
switch(opt)
 {
 case 1:
   c.lines();
   break;
 case 2:
   c.words();
   break;
 case 3:
   c.character();
   break;
 case 4:
   c.searchar();
   break;
 case 5:
   cout<<"\n\t\t Searching the pattern \n";
   cout<<"\t\t ----------------------";
   cout<<"\n\nEnter the Pattern to be searched \n";
   cin>>s;
   clrscr();
   temp = c.cpattern(s);
   if ( temp  == 0 )
     { cout<<" pattern not found \n"; }
   else
     { cout<<"\n\n\t\t Pattern to be Searched : "<<s;
       cout<<"\n\t\t  Number of occurrences  : "<<temp; }
   cout<<"\n press any key to continue \n\n";
   getch();
   break;
 case 6:
   c.replace();
   break;
 case 7:
   c.remove();
   break;
 case 8:
   c.insert();
   break;
 case 9:
   cout<<"\n\n\t\t Given Text is \n";
   cout<<"\t\t --------------";
   c.display();
```

```
      break;
     case 10:
      exit(0);
      break;
     }
    }
   while(1);
   }
```

# Ex 6.5

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class person
{
  private:
  int i, n, ro[10],m1[10],m2[10],m3[10],a[10],t[10];
  char na[25][10],c[25][10],r[10],g[10];
  public:
  void getdata();
  void printdata();
  void processdata();
  void p(int xx);
  void prindata();
  };
  int z, y, x, w, v, u = 0;

void person :: p(int xx)
 {
   for(i=1;i<=n;i++)
    {
     if( xx == ro[i] )
     {
cout << "\n\n================================================\n";
cout << "               students mark list             \n";
cout << "================================================\n";
cout << "   Name            : " << na[i] << endl;
cout << "   Roll no         : " << ro[i] << endl;
cout << "   Course          : " << c[i] << endl;
cout << "================================================\n";
cout << "   Sports          : " << m1[i] << endl;
cout << "   Communication   : " << m2[i] << endl;
cout << "   Performance     : " << m3[i] << endl;
cout << "================================================\n";
cout << "   Total           : " << t[i] << endl;
cout << "   Result          : " << r[i] << endl;
cout << "   Grade           : " << g[i] << endl;
cout << "================================================\n";
     }
     }
     }

void main()
  {
  clrscr();
   person p;
```

```
   p.getdata();
   getch();
   clrscr();
   p.printdata();
   getch();
   clrscr();
   p.processdata();
   while (1)
   {clrscr();
    cout << " \n\t\t\t MENU \n";
    cout << " \t\t\t======\n";
    cout << "\t|==============================|\n";
    cout << "\t| Mark List of all Students -1 |\n";
    cout << "\t| Mark List of Single one   -2 |\n";
    cout << "\t| Exit                      -3 |\n";
    cout << "\t|==============================|\n";
    cout << " Please, enter Ur, choice \t";
    int ch;
    cin >> ch;
    switch(ch)
    {
    case 1:
    clrscr();
   cout << "\n\t\t\t STUDENTS  MARK LIST \n";
  cout << "\t\t\t ====================\n\n";
  cout <<"-------------------------------------------------------";
  cout <<"---------------\n";
  cout <<"ROLLNO  NAME  COURSE \t\tMARKS\t\tTOTAL\tAVERAGE\tGRADE\tRESULT
\n";
  cout <<"-------------------------------------------------------";
  cout <<"---------------\n";
  cout <<"                        M1\tM2\tM3
\n";
  cout <<"-------------------------------------------------------";
  cout <<"---------------\n";
   p.prindata();
  cout <<"\n-------------------------------------------------------";
  cout <<"---------------\n";
  cout <<" \n grade A " << z;
  cout <<" \t grade B " << y;
  cout <<" \t grade C " << x;
  cout <<" \n grade D " << u;
  cout <<" \t grade E " << v;
  cout <<" \t failures " << w;
  cout << "\n\n";
  break;
case 2:
 clrscr();
 cout << " Enter the Student Roll Number \n";
 int xx;
 cin >> xx;
 p.p(xx);
 break;
case 3:
 goto ss;
 }
// cout << " press any key for continue \n";
```

```
 getch();
 }
ss:;
   }
   void person::getdata()
   {
   clrscr();
   cout << "\n Enter the total number of students ";
   cin >> n;
   cout << "\n Enter the information
name,rollno,mark1,mark2,mark3,course\n";
   for(i=1;i<=n;i++)
   {
     cin >> na[i] >> ro[i] >> m1[i] >> m2[i] >> m3[i] >> c[i];
     }
     }
void person::printdata()
    {
     cout << "\n students data list is ";
     for(i=1;i<=n;i++)
     {
      cout << "\n Name = \t"<< na[i] << "\n Rollno = \t"<< ro[i];
      cout << "\n Mark1 = \t" << m1[i] << "\n Mark2 = \t" << m2[i];
      cout << "\n Mark3 = \t" << m3[i] << "\n Course = \t" << c[i] <<endl;
      getch();
      }
      }

void person::processdata()
     {
       for(i=1;i<=n;i++)
       {
       t[i]=m1[i]+m2[i]+m3[i];
       a[i] = t[i]/3;
       if(m1[i] >= 50 && m2[i] >= 50 && m3[i] >= 50)
       { r[i]='p';
        if(a[i] >= 90)
        { g[i]='A';  z++; }
        else
        if(a[i] >= 80)
        {  g[i]='B'; y++; }
        else
        if(a[i] >= 70)
        { g[i]='C'; x++; }
        else
        if(a[i] >= 60)
        { g[i]='D'; u++; }
        else
        if(a[i] >= 50)
        { g[i]='E'; v++; }
        }
        else
        { r[i]='f';
        g[i]='-'; w++; }
      }
     }
    void person::prindata()
```

```
    {
      for(i=1;i<=n;i++)
      {
 cout <<"\n"<<ro[i]<<"\t"<<na[i]<<"\t"<<c[i]<<"\t"<<m1[i]<<"\t"<<m2[i];
 cout <<"\t"<<m3[i]<<"\t";
 cout <<t[i]<<"\t"<<a[i]<<"\t"<<g[i]<<"\t"<<r[i];
      }
      }
```

# Ex - 6.6

```
/ * Staff class */

#include <iostream.h>
#include <conio.h>
void main()
 {
class staff
{
private :

char name[25],date[11], qual[25];

public:
void get();
void display();
};
void staff :: get()
{
cout << " Enter the details of staff \n";
cout << " Enter the name, date of appointment and Qualification \n";
cin >> name >> date >> qual;
}

void student :: display()
{
cout << " Staff Details \n";
cout << " Name  " << name << endl;
cout << " Date of Appointment " << date << endl;
cout << " Qualification " << qual << endl;
}

void main()
{
staff s;
s.get();
s.display();

}
```

# Ex – 6.7

```
#include<iostream.h>
#include<conio.h>
class room
```

```
{
       float length, breadth, height, farea, warea;
       int  fans, window, door;
public:
room(float l, float b, float h, float w, float f, int f1, int w1, int d )
       {
       length = l;
       breadth = b;
       height = h;
       farea = f;
       warea = w;
       fans = f1;
       window = w1;
       door = d;
       }
void display()
  {
       cout << " The Room details " << endl;
       cout << length << "\n" << breadth << "\n" << height << "\n"
            << farea<<"\n" << warea;
       cout << "\n"<<fans <<"\n"<< window <<"\n"<< door<<endl;
     }
    };
void main()
{
   room r(12.3,23.2,34.1,23.2,24.4,5,6,4);
   r.display();
   getch();
}
```

**Lab Unit – 7 (2 Hrs Real Time)**

**Ex - 7.1**

```
# include <iostream.h >
class time
{
private :
   int h, m, s;
public :
       time()
           { h = m = s = 0; }
       time(int h1, int m1, int s1)
           { h = h1 ; m = m1; s= s1;  }
       void display()
           { cout << n << ":" << m << ":" << s; }
       void add(time t1, time t2)
             {
                 s = t1.s + t2.s;
                 if (s > 59)
                   { s -= 60; m++; }
                    m += t1.m + t2.m;
                    if ( m > 59)
                  { m -= 60; h++;}
                      h += t1.h + t2.h;
```

```
                }
        };
void main()
    {
         time t1(5,59,59);
        time t2(4,30,30);
        time t3;
        t3.add(t1,t2);
        cout << " \n time = ";
        t3.display();
     }
```

**Ex - 7.2**

```
        /*       VARIOUS COMPLEX NUMBER OPERATIONS
                 < USING CONSTRUCTOR OVERLOADING >          */

#include<iostream.h>
#include<conio.h>
class complex
{
 private:
  float a, b;
 public:
  complex() { }
  complex(float r, float i)
  { a = r; b = i; }
  void menu();
  friend complex c(complex, complex);
  friend complex su(complex, complex);
  friend complex di(complex, complex);
  friend complex mu(complex, complex);
  friend void s(complex);
  };
void complex :: menu()
  {
   cout << "\n\n\t\t|==============================|\n";
   cout << "\t\t| Operations of Complex Number |\n";
   cout << "\t\t|==============================|\n";
   cout << "\t\t|  ADDITION        - 1         |\n";
   cout << "\t\t|  SUBTRACTION     - 2         |\n";
   cout << "\t\t|  MULTIPLICATION  - 3         |\n";
   cout << "\t\t|  DIVISION        - 4         |\n";
   cout << "\t\t|  EXIT            - 5         |\n";
   cout << "\t\t|==============================|\n\n";
   cout << "\n\t\t Please, Enter Any Choice \t";
   }
  complex c(complex c1,complex c2)
  { complex c3;
    c3.a=c1.a+c2.a;
    c3.b=c1.b+c2.b;
    return(c3);
    }
 complex su(complex w1,complex w2)
    { complex w3;
      w3.a=w1.a-w2.a;
      w3.b=w1.b-w2.b;
      return(w3);
```

```
   }
 complex mu(complex u1,complex u2)
   { complex u3;
    u3.a=u1.a * u2.a + u1.b * u2.b;
    u3.b=u1.a * u2.b + u1.b * u2.a;
    return(u3);
    }
 complex di(complex u1,complex u2)
 {
  int t;
  complex u3;
  t=(u1.a * u2.a) + (u1.b * u2.b);
  u3.a=(u1.a * u2.a) + (u1.b * u2.b)/t;
  u3.b=((u2.a * u1.b) - (u2.a * u1.b)/t);
  return(u3);
  }
 void s(complex c)
 {
 if(c.b < 0)
 cout << c.a << " - i" << (-c.b) << "\n";
 else
 cout << c.a << " + i" << c.b << "\n";
 }
void main()
{
  clrscr();
  int ch;
  float f, g, h, j;
  cout << " Please, Enter The First Complex Number \n";
  cin >> f >> g;
  complex x(f, g);
  cout << " Please, Enter The Second Complex Number \n";
  cin >> h >>j;
  complex y(h, j);
  complex z;
  z = c(x, y);
  while (1)
  {
  clrscr();
  cout << " \n The First Complex Number is " ;
  s(x);
  cout << " \n The Second Complex Number is ";
  s(y);
  z.menu();
  cin  >> ch;
  cout << "\n";
  switch(ch)
   {

    case 1:
     cout << " \n The Addition of The Complex Numbers is ";
     complex z;
     z = c(x, y);
     s(z);
     break;

    case 2:
```

```
        complex z1;
        z1=su(x, y);
        cout << " \n the subtraction of the complex numbers is ";
        s(z1);
        break;

       case 3:
        complex z2;
        z2=mu(x, y);
      cout << " \n the multiplication of the complex numbers is ";
        s(z2);
        break;

        case 4:
        complex z3;
        z3=di(x, y);
        cout << "\n The Division of The Complex Numbers is ";
        s(z3);
        break;

        case 5:
         goto end;
         break;
        }
        cout << "\n Press Any Key To Continue";
        getch();
        }
end:;
}
```

**Ex - 7.3**

```
 // conversion of measurements feets to inches Centimeter to meter

#include <iostream.h>
#include <conio.h>
class db;
class dm
  {  int m;
     int cm;
     public:
      void getdata();
      void printdata();
      friend void add(dm, db, int);
  };
class db
  {  int feet;
     int inches;
     public:
      void getdata();
      void printdata();
      friend void  add(dm, db, int);
  };

void dm::getdata()
 { cout << "\n Enter the distance in Meters and centimeters \n";
```

```
   cin >> m >> cm;
 }
void dm::printdata()
 { cout << "\n Distance in centimeters and meters \n";
   cout << " ================================= \n";
   cout << "========================\n";
   cout << " Meters      = \t " << m;
   cout << "\n Centimeters = \t " << cm << "\n";
   cout << "========================\n";
 }

void db::getdata()
 { cout << "\n Enter the distance in feets and inches \n";
   cin >> feet >> inches;
 }
void db::printdata()
 { cout << "\n Distance in Feets and inches \n";
   cout << " =========================== \n";
   cout << "===================\n";
   cout << " Feets   = \t " << feet;
   cout << "\n Inches  = \t " << inches << "\n";
   cout << "===================\n";
 }

void add(dm d1,db d2,int opt)
 { dm c;
   db b;
   if ( opt == 1 )
     {  b.inches  = d2.feet * 12 + d2.inches + (d1.m * 100 + d1.cm) / 2.5;
       b.feet =  b.inches / 12;
       b.inches = b.inches % 12;
       b.printdata();
      }
   else
    {
       c.cm = d1.m * 100 + d1.cm + (d2.feet * 12 + d2.inches) * 2.5;
       c.m =  c.cm / 100;
       c.cm = c.cm % 100;
       c.printdata();
    }
 }

 void main()
  {  int opt;
     dm d1;
     clrscr();
     d1.getdata();
     db d2;
     d2.getdata();
     clrscr();
     d1.printdata();
     d2.printdata();
     getch();
     clrscr();
as:  cout << "\t Enter the units in which you want the output " << endl;
     cout << "\t ===================================== \n\n";
     cout << "\t\t |============================|\n";
```

```
        cout << "\t\t | 1. Feets and Inches         |" << endl;
        cout << "\t\t | 2. centimeters  and feet    |" << endl;
        cout << "\t\t |============================|\n";
        cout << " \n\n\t\t Please, Enter Ur,chioce \t";
        cin >> opt;
        cout << "\n";
        cout << "\n  Addition of two Distance \t ";
        add(d1,d2,opt);
        cout << "\n use [1] to continue \t";
        cout << "\n";
        int v;
        cin >> v;
        if( v== 1)
        goto as;
      }
```

**Ex - 7.4**

```
      /*  USER - DEFINED MANIPULATIONS                    */

#include<iostream.h>

#include<conio.h>
#include<iomanip.h>
int i = 1;
ostream & form2(ostream & output);    //ITEM
ostream & form1(ostream & output);    //CODE
ostream & form(ostream & output);      //NAME
ostream & form3(ostream & output);    //COST,PRICE
ostream & form4(ostream & output);     //QUANTITY
class shop

 {
   private:
        int item[5];
        int qu[3];
        float price[8];
        float cost[8];
        char name[20][20];
   public:
        void get();
        void print();
 };
void shop :: get()
 {
   clrscr();
   getch();
   cout << "\n";
   cout<<"\t\tEnter the item code \n\t\t";
   cin >> item[i];
   cout<<"\t\tEnter the item name \n\t\t";
   cin >> name[i];
   cout<<"\t\tEnter the quantity of an item \n\t\t";
   cin >> qu[i];
   cout<<"\t\tEnter the price of an item \n\t\t";
   cin >> price[i];
```

```
   cost[i] = qu[i] * price[i];
   i++;
 }
void shop :: print()
 {
  int j;
  clrscr();
  cout<<".................................................\n";
  cout<<"Code";
  cout<<"\t       Name";
  cout<<"\t       Quantity";
  cout<<"\t       Price";
  cout<<"\t       Cost";
  cout<<"\n";
  cout<<".................................................\n";
  for(j=1;j<i;j++)

   {
    cout << form1;
    cout << item[j];
    cout <<" ";
    cout<< form2;
    cout << name[j];
    cout<<"\t\t";
    cout << form4 << qu[j];
    cout<<"\t\t";
    cout<< form3 << price[j];
    cout<<"\t\t";
    cout<< form3 <<cost[j];
    cout<<"\n";
   }
cout<<".................................................\n";
 }
ostream & form1(ostream & output)       //CODE
{
 output.width(5);
 output.setf(ios::right, ios::adjustfield);
 output.fill('.');
 return output;
}

ostream & form2(ostream & output)           //ITEM
{
 output.setf(ios::left, ios::adjustfield);
 output.width(8);
 return output;
}

ostream & form3(ostream & output)      //COST,PRICE
{
 output.setf(ios::showpoint);
 output.precision(2);
 output.width(6);
 output.fill('.');
 output.setf(ios::right, ios::adjustfield);
 return output;
}
```

```
ostream & form4(ostream & output) //QUANTITY
{
 output.width(3);
 output.fill('.');
 return output;
}

void main()
 {
  shop s;
  int f=1;
  int ch;
  while(f==1)
  {
   clrscr();
   cout<<"\t\t\t   Menu \n";
   cout<<"\t\t\t   ~~~~ \n\n";
   cout<<"\t\t 1 -> Enter an Item \n";
   cout<<"\t\t 2 -> Display the List \n";
   cout<<"\t\nEnter your choice";
   cin >> ch;
   switch(ch)
   {
    case 1:
          s.get();
          break;
    case 2:
          s.print();
          break;
   }
 getch();
 clrscr();
 cout << "\t\n Do you want continue \n";
 cin >> f;
 }
getch();
}
```

# Ex - 7.5

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
int i = 1;
class shop
 {
   private:
         int item[5];
         int qu[3];
         float price[8];
         float cost[8];
         char name[20][20];
   public:
         void get();
```

```
        void print();
 };
void shop :: get()
 {
   clrscr();
   getch();
   cout<<"\nEnter the item code \n";
   cin >> item[i];
   cout<<"Enter the item name \n";
   cin >> name[i];
   cout<<"Enter the quantity of an item \n";
   cin >> qu[i];
   cout<<"Eenter the price of an item \n";
   cin >> price[i];
   cost[i] = qu[i] * price[i];
   i++;
 }
void shop :: print()
 {
  int j;
  clrscr();
  cout<<".................................................\n";
  cout<<"Code";
  cout<<"\t\tName";
  cout<<"\tQuantity";
  cout <<"\t Price";
  cout <<"\t\t Cost";
  cout<<"\n";
  cout<<".................................................\n";
  for(j=1;j<i;j++)
   {
    cout<<setiosflags(ios::right);
    cout << item[j];
    cout <<"\t";
    cout<<setiosflags(ios::left);
    cout << name[j];
    cout.setf(ios::right, ios::adjustfield);
    cout<<"\t\t";
    cout << qu[j];
    cout.setf(ios::right, ios::adjustfield);
    cout<<"\t\t";
    cout.precision(2);
    cout.setf(ios::showpoint);
    cout<<price[j];
    cout.setf(ios::right, ios::adjustfield);
    cout<<"\t\t";
    cout.precision(2);
    cout.setf(ios::showpoint);
    cout<<cost[j];
    cout<<"\n";
   }
cout<<".................................................\n";
 }
void main()
 {
  shop s;
  int f=1;
```

```
  int ch;
  while(f==1)
  {
   clrscr();
   cout<<"\t Menu \n";
   cout<<"\t1.Enter an item \n";
   cout<<"\t2.Print the item \n";
   cout<<"\t\nEnter your choice";
   cin >> ch;
   switch(ch)
   {
    case 1:
          s.get();
          break;
    case 2:
          s.print();
          break;
   }
 cout << "\t\n Do you want continue \n";
 cin >> f;
 }
getch();
}
```

**Lab Unit - 8 (2 Hrs Real Time)**

**Ex - 8.1**

```
#include<iostream.h>
#include<iomanip.h>

class a
             {
              private:
                    char name[15];
                     int num;
               public:
                    void getdata();
                     void display();
            };

class b:public a
        {
              private:
                     int tele_num;
               public:
                    void getdata();
                     void display();
           };

void a::getdata()
          {
             cout<<"Enter the name   ";
             cin>>name;
             cout<<"Enter the Register number ";
```

```
            cin>>num;
            }

void a::display()
          {
            cout<<name;
            cout<<num;
            }

void b::getdata()
          {
             a::getdata();
            cout<<"Enter the telephone number   ";
            cin>>tele_num;
           }

void b::display()
          {
      a::display();
            cout<<tele_num;
           }

void main()
         {
               b enter;
               enter.getdata();
               enter.display();
         }
```

## Ex 8.2

```
#include<iostream.h>
#include<iomanip.h>

class a
{
             private:
                   char name[15];
                    int num;
              public:
                   void getdata();
                    void display();
           };


class b
        {
              private:
                    int tele_num;
              public:
                   void getdata();
                    void display();
           };

class c : public a, public b
        {
              private:
```

```
                    int city_num;
                public:
                    void getdata();
                     void display();
              };


void a::getdata()
          {
             cout<<"Enter the name   ";
             cin>>name;
             cout<<"Enter the Register number ";
             cin>>num;
             }

void a::display()
          {
             cout<<name;
             cout<<num;
             }

void b::getdata()
          {
       cout<<"Enter the telephone number   ";
             cin>>tele_num;
            }

void b::display()
          {
                    cout<<tele_num;
            }

void c::getdata()
          {
               a::getdata();
               b::getdata();
       cout<<"Enter the city pincode number   ";
             cin>>city_num;
            }
void c::display()
          {
           a::display();
            b::display();
           cout<<city_num;
           }

void main()
          {
                c enter;
                enter.getdata();
                enter.display();
          }
```

# Ex – 8.3

```
#include<iostream.h>
```

```
#include<iomanip.h>

class a
{
              private:
                    char name[15];
                     int num;
               public:
                    void getdata();
                     void display();
           };

class b:public a
        {
              private:
                     int tele_num;
               public:
                    void getdata();
                     void display();
           };

class c : public b
        {
              private:
                     int city_num;
               public:
                    void getdata();
                     void display();
           };

void a::getdata()
          {
             cout<<"Enter the name   ";
             cin>>name;
             cout<<"Enter the Register number ";
             cin>>num;
             }

void a::display()
          {
             cout<<name;
             cout<<num;
             }


void b::getdata()
          {
       a::getdata();
       cout<<"Enter the telephone number   ";
             cin>>tele_num;
           }
void b::display()
          {
                    a::display();
                   cout<<tele_num;
           }
```

```
void c::getdata()
          {

               b::getdata();
       cout<<"Enter the city pincode number   ";
             cin>>city_num;
            }
void c::display()
          {

            b::display();
           cout<<city_num;
            }
void main()
          {
                c enter;
                enter.getdata();
                enter.display();

          }
```

## Ex - 8.4

```
#include<iostream.h>
#include<iomanip.h>

class a
{
              private:
                    char name[15];
                     int num;
               public:
                    void getdata();
                     void display();
            };

class b
        {
              private:
                     int tele_num;
               public:
                    void getdata();
                     void display();
            };
class c : public a, public b
        {
              private:
                     int city_num;
               public:
                    void getdata();
                     void display();
            };

class d: public c
        {
```

```
              private:
                     int car_num;
               public:
                    void getdata();
                     void display();
            };

void a::getdata()
          {
             cout<<"Enter the name   ";
             cin>>name;
             cout<<"Enter the Register number ";
             cin>>num;
             }

void a::display()
          {
             cout<<name;
             cout<<num;
             }

void b::getdata()
          {
       cout<<"Enter the telephone number   ";
             cin>>tele_num;
            }
void b::display()
          {
                    cout<<tele_num;
            }

void c::getdata()
          {
               a::getdata();
               b::getdata();
       cout<<"Enter the city pincode number   ";
             cin>>city_num;
            }

void c::display()
          {
           a::display();
            b::display();
           cout<<city_num;
            }

void d::getdata()
          {
               c::getdata();
               cout<<"Enter the car number   ";
             cin>>car_num;
            }
void c::display()
          {
           c::display();

           cout<<car_num;
```

```
            }

void main()
          {
                d enter;
                enter.getdata();
                enter.display();
          }
```

# Ex 8.5

```
#include<iostream.h>
#include<iomanip.h>

class a
{
              private:
                    char name[15];
                     int num;
               public:
                    void getdata();
                     void display();
            };


class b:public a
        {
              private:
                     int tele_num;
               public:
                    void getdata();
                     void display();
            };

class c : public a
        {
              private:
                     int city_num;
               public:
                    void getdata();
                     void display();
            };


void a::getdata()
          {
             cout<<"Enter the name   ";
             cin>>name;
             cout<<"Enter the Register number ";
             cin>>num;
             }

void a::display()
          {
             cout<<name;
             cout<<num;
```

```
            }

void b::getdata()
          {
             a::getdata();
       cout<<"Enter the telephone number   ";
             cin>>tele_num;
           }

void b::display()
          {
             a::getdata();
                   cout<<tele_num;
           }

void c::getdata()
          {
               a::getdata();
             cout<<"Enter the city pincode number   ";
             cin>>city_num;
           }

void c::display()
          {
           a::display();
          cout<<city_num;
           }

void main()
          {
                c enter;
                enter.getdata();
                enter.display();
          }
```

# Ex – 8.6

```
Display the University details              */

#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <process.h>
#include <iomanip.h>
class univer
{  protected:
   char na[20],p[20];
   public:
    void get()
    {
     cout << " enter name and place \n";
     cin >> na >> p;
     }
    void dis()
    {
     cout << "\n\t\t NAME OF THE UNIVERSITY    : " << na
         << "\n\t\t PLACE OF THE UNIVERSITY   : " << p <<"\n";
```

```
     }
    };
class depart : protected univer
{
  char dn[20];
  int ns,sn;
  public:
   void ge()
   {
    get();
    cout << " enter the dept name,no of students,no of staff \n";
    cin >> dn >> ns >> sn;
    }
   void disp()
   {
   dis();
   cout   << "\t\t DEPARTMENT NAME           : " << dn
         << "\n\t\t NO OF STUDENTS            : " << ns
         << "\n\t\t NO OF STAFF               : " << sn;
   cout << "\n";
   }
   };
 class staff : private depart
 {
   public:
   char nas[20],su[20];
   int sa,ex;
   void g()
   { ge();
     cout << "enter the name,salary,experience,&subject\n";
     cin >> nas >> sa >> ex >> su;
     }
   void d()
   {getch();
    clrscr();
    cout  << "\n\t\t   STAFF DETAILS OF UNIVERSITY \n"
         << "\t\t -------------------------------\n";
    disp();
    cout  << "\t\t NAME OF THE STAFF         : " << nas
         << "\n\t\t SALARY                    : " << sa
         << "\n\t\t EXPERIENCE                : " << ex
         << "\n\t\t SUBJECT                   : " << su <<endl;
    }
   };
class stud : private depart
{
 public:
 char nsa[20],c[20];
 int yc,to;
 void g1()
 {ge();
  cout << " enter the name,year,course & total marks of the student \n";
  cin >> nsa >> c >> yc >> to;
  }
 void d1()
 {   getch();
     clrscr();
```

```
     cout << "\n\t\t  STUDENTS DETAILS OF THE UNVERSITY \n"
        << "\t\t ---------------------------------------\n";
     disp();
     cout << "\t\t NAME OF THE STUDENT       : " << nsa
        << "\n\t\t COURSE                    : " << c
        << "\n\t\t YEAR OF COURSE            : " << yc
        << "\n\t\t TOTAL MARKS               : " << to << "\n";
 }
};
void main()
{
clrscr();
staff sg;
stud st;
int ch;
while(1)
{ getch();
  clrscr();
  cout << " \n\t\t\t menu\n";
  cout << " \n\t\t\t++++++\n";
  cout << " \t\tStaff information   -1 \n";
  cout << " \t\tStudent information -2 \n";
  cout << " \t\tExit                -3 \n";
  cout << " \t\tenter your choice \n";
  cin >> ch;
  switch(ch)
  {
  case 2:
  st.g1();
  st.d1();
  break;
  case 1:
  sg.g();
  sg.d();
  break;
  case 3:
  exit(0);
  break;
  }
  }
  getch();
  }
```

## Ex - 8.7

```
*/              PERSONS DETAILS                          */


# include <iostream.h>
# include <conio.h>

class person
{
   protected :
       char name[20];
       char date[11];
       char sex;
```

```
        float h,w;
    public :
        virtual void getd()
        {
           cout << "\n\tThe Name Of Person : ";
           cin >> name;
           cout << "\n\tThe Date Of Birth : ";
           cin >> date;
           cout << "\n\tThe Sex : ";
           cin >> sex;
           cout << "\n\tThe Height Of Person : ";
           cin >> h;
           cout << "\n\tThe Weight Of Person : ";
           cin >> w;
        }

        virtual void putd()
        {
           cout << "\n\t\t\tPerson Details";
           cout << "\n\t\t\t~~~~~~~~~~~~~~~";
           cout << "\n\n\tName Of Person : ";
           cout << name;
           cout << "\n\n\tDate Of Birth : ";
           cout << date;
           cout << "\t\t\tSex : ";
           cout << sex;
           cout << "\n\n\tHeight Of Person : ";
           cout << h;
           cout << "\t\t\tWeight Of Person : ";
           cout << w;
        }
};
class major : public person
{
   protected :
        char status;
        char job[20];
   public :
        void getd()
        {
 cout << "\n\tStatus Of Person [m-married/u-unmarried/w-widow]: ";
         cin >> status;
         cout << "\n\tJob Of Person : ";
         cin >> job;
        }
};
class minor : public person
{
    protected :
        char fname[20];
    public :
        void getd()
        {
            cout << " \n\tName Of father : ";
            cin >> fname;
        }
};
```

```
class other : public person
{
    protected :
      char hobby[20];
    public :
      void getd()
      {
          cout << " \n\tThe Hobby Of Person : ";
          cin >> hobby;
      }
      void putd()
      {
          cout << " \n\tHobby Of Person : ";
          cout << hobby;
      }
};

void main()
{
     major m1;
     minor m2;
     other o1;
     int ch;
     person p,*pt;
     pt=&p;
     pt->getd();
     pt=&o1;
     pt->getd();
     clrscr();
     pt=&p;
     pt->putd();
     cout << " \n\n\t1.Major | 2.Minor [1/2] : ";
     cin >>ch;
     switch(ch)
     {
        case 1:
              pt=&m1;
              pt->getd();
              break;
        case 2:
              pt=&m2;
              pt->getd();
              break;
     }
     pt=&o1;
     pt->putd();
     getch();
}
```

# Ex - 8.8

```
checking the patient condition              */
#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <process.h>
#include <iomanip.h>
```

```
class patient
{
   char na[20],s[7];
   int a;
   public:
    void get()
    {
     cout << " enter name and sex & age \n";
     cin >> na >> s >> a;
     }
    void dis()
    {
     clrscr();
     cout << "\n\n\t\t\t  MEENA HOSPITAL   \n"
         << "\t\t\t ================   \n"
         << "\n\t\t\t PATIENTS DESCRIPTION "
         << "\n\t\t\t=====================\n";
     cout << "\n\n\t\t PATIENT NAME              : " << na
         << "\n\t\t SEX                       : " << s
         << "\n\t\t AGE                       : " << a << "\n";
     }
    };
class descript : virtual public patient
{
  char ni[20],da[11];
  int wn,bn;
  public:
   void ge()
   {
    get();
   cout << " enter the wardno,bedno,nameofill & date of admission \n";
    cin >> wn >> bn >> ni >> da;
    }
   void disp()
   {dis();
     cout << "\t\t WARD NUMBER              : " << wn
         << "\n\t\t BED NUMBER               : " << bn
         << "\n\t\t NAME OF ILLNESS          : " << ni
         << "\n\t\t DATE OF ADMISSION        : " << da;
   cout << "\n";
   }
   };
 class doctor : virtual public patient
 {
   public:
   char nad[20],qu[20];
   void g()
   {
     cout << "enter the name,qualification\n";
     cin >> nad >> qu;
     }
   void d()
   {
     cout << "\t\t NAME OF THE DOCTOR        : " << nad
         << "\n\t\t QUALIFICATION OF DOCTOR   : " << qu << "\n";
    }
   };
```

```
class cond : public doctor,public descript
{
 public:
 char c[20];
 void g1()
 {ge();
  g();
  cout << "\n\t ENTER THE CONDITION OF PATIENT \n";
  cin >> c;
  }
 void d1()
 {disp();
  d();
    cout  << "\t\t CONDITION OF PATIENT      : " << c << "\n";
 }
};
void main()
{
clrscr();
cond pt;
  pt.g1();
  pt.d1();
  getch();
  }
```

## Ex – 8.9

```
#include <iostream.h>
#include <conio.h>
#include < string.h>
#include <stdio.h>
#include <process.h>
class player
 {  protected :
     char name[20];
     char nation[20];
    public:
       void getdata();
       virtual void putdata() {}
  };
void player::getdata()
 {  cout<<"\n\t Enter the player name    :  "; cin>>name;
    name[strlen(name)] = '\0';
    cout<<"\n\t Enter the player nation  :  "; cin>>nation;
    nation[strlen(nation)] = '\0';
 }
class event : public player
  {
     char *etype,*ename;
    public:
      void indata()
       { cout<<"\n\t\t Enter Event type :  ";cin>>etype;
        cout<<"\n\t\t Enter Event name :  ";cin>>ename;
       }
      void putdata();
  };
```

```
void event :: putdata()
       {  cout<<"\n\t\t Player Name  : "<<name;
         cout<<"\n\t\t Player Nation : "<<nation;
         cout<<"\n\t\t Event Type    : "<<etype;
         cout<<"\n\t\t Event Name    : "<<ename;
       }

class personnel:public player
 {  float ht,wt;
    int age;
    public:
      void getdata1()
       {  cout<<"\n\t\t Enter height :  ";cin>>ht;
         cout<<"\n\t\t Enter Weight  :  ";cin>>wt;
         cout<<"\n\t\t Enter Age     :  ";cin>>age;
       }
      void putdata()
       { cout<<"\n\t\t Player Name  : "<<name;
        cout<<"\n\t\t Player Nation : "<<nation;
        cout<<"\n\t\t    Height     : "<<ht;
        cout<<"\n\t\t    Weight     : "<<wt;
        cout<<"\n\t\t     Age       : "<<age;
       }
 };

void main()
 {  int opt,t;
    player pl; player *plptr;
    event e;
    personnel p;
    clrscr();
    cout<<"\n\t\t Entering the details of the player ";
    cout<<"\n\t\t -------------------------------- ";
    pl.getdata();
    e.indata();
    p.getdata1();
    char q;
    q = 'y';
    while( q  ==  'y')
     {  clrscr();
       cout<<"\n\n\t\t\t  Menu";
       cout<<"\n\t\t\t ------ \n";
       cout<<"\n\t\t 1. Event Details of player ";
       cout<<"\n\t\t 2. Personnal Details of pLayer ";
       cout<<"\n\t\t 3. Exit  ";
       cout<<"\n\n Enter your Choice   ";
       cin>>opt;
       clrscr();
       switch(opt)
        { case 1:
            cout<<"\n\n\t\t Event Details of the Player ";
            cout<<"\n\t\t ------------------------- ";
            plptr = &e;
            plptr->putdata();
            getch();
            break;
          case 2:
```

```
            cout<<"\n\n\t\t Personnel Details of the Player ";
            cout<<"\n\t\t---------------------------------";
            plptr = &p;
            plptr->putdata();
            getch();
            break;
          case 3:
            q = 'n';
            break;
         }
       }
      getch();
      exit(1);
    }
```

# Ex - 8.10

```
#include<iostream.h>
#include<string.h>
#include<iomanip.h>
#include<process.h>
#include<stdio.h>
#include<conio.h>
int i = 0;
class details
 { public:
      char title[10][10],author[10][10];
      int code[10];
 };
class publications
  {  public:
      int year[10];
      char pname[10][10];
  };
class derived:public details,public publications
 {  public:
      float price[10];
      char available[3][10];
      void newbook();
      void editbook();
      void listbook();
      void availablebook();
      void sortbook();
  };
void derived::newbook()
 {  cout<<"\n\t\t Adding a New Book ";
    cout<<"\n\t\t ----------------- ";
    cout<<" \n\t Enter Book Name           :  ";cin>>title[i];
    cout<<" \n\t Enter Book Code           :  ";cin>>code[i];
    cout<<" \n\t Enter Author Name         :  ";cin>>author[i];
    cout<<" \n\t Enter Year of Publication :  ";cin>>year[i];
    cout<<" \n\t Enter Publisher Name      :  ";cin>>pname[i];
    cout<<" \n\t Enter Price of Book       :  ";cin>>price[i];
    cout<<" \n\t Enter the Availability    :  ";cin>>available[i];
    i++;
 }
void derived::editbook()
```

```
{ char y = 'y',temp[12];
  int j,opt,c,t;
  float t1;
  cout<<"\n\n\t\t Editing A Book Details ";
  cout<<"\n\t\t ----------------------";
  cout<<"\n\n Enter the code of the Book to be Edited  : ";
  cin>>c;
  int g= 1;
  for(j = 0; j < i;j++)
   if ( c == code[j] )
    { g = 0;
      while ( y == 'y' || y == 'Y')
      {  clrscr();
         cout<<"\n\n\t\t Menu For Edition ";
         cout<<"\n\t\t----------------- \n";
         cout<<"\n\t\t 1. Book Name ";
         cout<<"\n\t\t 2. Author ";
         cout<<"\n\t\t 3. Year of Publications ";
         cout<<"\n\t\t 4. Publisher Name  ";
         cout<<"\n\t\t 5. Price ";
         cout<<"\n\t\t 6. Availability ";
         cout<<"\n\t\t 7. Exit ";
         cout<<"\n\n Enter your Choice :  ";cin>>opt;clrscr();
         switch(opt)

         { case 1:
            cout<<"\n\n\t\t Editing Book Name ";
            cout<<"\n\t\t -----------------";
            strcpy(temp,title[j]);
            cout<<"\n\n\t\t Current Book Name :  "<<title[j];
    cout<<"\n\n Enter the New Book  Name  : ";cin>>title[j];
            clrscr();
            cout<<"\n\n\t\t Editing Book Name ";
            cout<<"\n\t\t -----------------";
            cout<<"\n\n\t\t Old Book Name :  "<<temp;
            cout<<"\n\n\t\t New Book Name :  "<<title[j];
            cout<<"\n\n";
            getch(); break;

           case 2:
            cout<<"\n\n\t\t Editing Book Author ";
            cout<<"\n\t\t -------------------";
            strcpy(temp,author[j]);
            cout<<"\n\n\t\t Current Book Author  :  "<<author[j];
    cout<<"\n\n Enter the New Book Author   : ";cin>>author[j];
            clrscr();
            cout<<"\n\n\t\t Editing Book Author ";
            cout<<"\n\t\t -------------------";
            cout<<"\n\n\t\t Old Book Author  :  "<<temp;
            cout<<"\n\n\t\t New Book Author :  "<<author[j];
            cout<<"\n\n";
            getch();   break;
           case 3:
            cout<<"\n\n\t\t Editing Publication Year ";
            cout<<"\n\t\t --------------------------";
            t = year[j];
    cout<<"\n\n\t\t Current Publication Year :  "<<year[j];
```

```
            cout<<"\n\n Enter the New Year of Publication  : ";
            cin>>year[j];
            clrscr();
            cout<<"\n\n\t\t Editing Publication Year ";
            cout<<"\n\t\t ---------------------------";
      cout<<"\n\n\t\t Old value of Publication year :  "<<t;
      cout<<"\n\n\t\t New value of Publication year :  "<<year[j];
            cout<<"\n\n";
            getch(); break;

           case 4:
            cout<<"\n\n\t\t Editing Book Publisher ";
            cout<<"\n\t\t ------------------------";
            strcpy(temp,pname[j]);
      cout<<"\n\n\t\t Current Book Publisher  :  "<<pname[j];
      cout<<"\n\n Enter the New Publisher  : ";cin>>pname[j];
            clrscr();
            cout<<"\n\n\t\t Editing Book Publisher ";
            cout<<"\n\t\t  ------------------------";
            cout<<"\n\n\t\t Old Publisher  :  "<<temp;
            cout<<"\n\n\t\t New Publisher  :  "<<pname[j];
            cout<<"\n\n";
            getch(); break;


           case 5:
            cout<<"\n\n\t\t  Editing Price ";
            cout<<"\n\t\t -----------------";
            t1 = price[j];
            cout<<"\n\n\t\t Current Price  :  "<<price[j];
            cout<<"\n\n Enter the New Pricen  : ";
            cin>>price[j];
            clrscr();
            cout<<"\n\n\t\t Editing Price ";
            cout<<"\n\t\t ----------------";
            cout<<"\n\n\t\t Old Price :  "<<t1;
            cout<<"\n\n\t\t New Price :  "<<price[j];
            cout<<"\n\n"; break;
            getch();
           case 6:
            cout<<"\n\n\t\t Editing the Availability ";
            cout<<"\n\t\t  -------------------------";
            strcpy(temp,available[j]);
      cout<<"\n\n\t\t Current Status of Availability  :  "<<temp;
            cout<<"\n\n Enter the New Status of Availability : ";
            cin>>available[j];
            clrscr();
            cout<<"\n\n\t\t Editing the Availability ";
            cout<<"\n\t\t --------------------------";
      cout<<"\n\n\t\t Old Status of Availability  :  "<<temp;
   cout<<"\n\n\t\t New Status of Availability  :  "<<available[j];
            cout<<"\n\n";
            getch(); break;
          default:
            y = 'n';
            break;
         }
```

```
          }
        }
        if ( g == 1 )
          { cout<<"\n\t\t Book code invalid ";
            getch();}
        }

void derived::listbook()
 { clrscr();
   cout<<"\n\n\t\t List of All Books ";
   cout<<"\n\t\t ------------------";
   cout<<"\n\n--------------------";
   cout<<"--------------------";
 cout<<"\n Code       Title      Author     Year   Publisher    ";
 cout<<"  Price       Availability";
   cout<<"\n-----------------";
   cout<<"-----------------";
   for ( int j = 0; j < i ;j++)
    { cout<<"\n"<<setw(4)<<code[j]<<setw(14)<<title[j];
      cout<<setw(14)<<author[j]<<setw(6)<<year[j]<<setw(12)<<pname[j];
      cout<<setw(10)<<price[j]<<setw(15)<<available[j];
    }
   cout<<"\n-----------------";
   cout<<"-----------------";
   getch();
  }
void derived::availablebook()
 { clrscr();
   cout<<"\n\n\t\t List of Available Books ";
   cout<<"\n\t\t ------------------------";
   cout<<"\n\n---------------------------";
   cout<<"--------------------";
 cout<<"\n Code       Title      Author     Year   Publisher    ";
   cout<<"  Price       Availability";
   cout<<"\n----------------------";
   cout<<"-----------------";
   for ( int j = 0; j < i ;j++)
     if ( strcmp(available[j],"yes") ==  0 )
     { cout<<"\n"<<setw(4)<<code[j]<<setw(14)<<title[j];
       cout<<setw(14)<<author[j]<<setw(6)<<year[j]<<setw(12)<<pname[j];
       cout<<setw(10)<<price[j]<<setw(15)<<available[j];
     }
   cout<<"\n----------------------------";
   cout<<"-----------------";
    getch();
  }

void derived::sortbook()
 { int j,k,t;char temp[10];float t1;
   clrscr();
   cout<<"\n\n\t\t Sorting Based on Book title ";
   cout<<"\n\t\t----------------------------";
   for( j = 0; j < i - 1;j++)
    for(k = j + 1; k < i; k++)
    if ( strcmp(title[j],title[k]) > 0 )
    {strcpy(temp,title[j]);strcpy(title[j],title[k]);strcpy(title[k],temp);
     strcpy(temp,author[j]);strcpy(author[j],author[k]);
```

```
        strcpy(author[k],temp);
        strcpy(temp,pname[j]);strcpy(pname[j],pname[k]);strcpy(pname[k],temp);
        t1 = price[j];price[j] = price[k];price[k] = t1;
        t =  code[j];code[j] = code[k];code[k] = t;
        t = year[j];year[j] = year[k];year[k] = t;
        strcpy(temp,available[j]);strcpy(available[j],available[k]);
        strcpy(available[k],temp);
       }
     }
void main()
 { int ch;
   derived d;
   while (1)
   { clrscr();
     cout<<"\n\n\t\t\t Menu ";
     cout<<"\n\t\t\t------";
     cout<<"\n\t\t 1. Entering New Books ";
     cout<<"\n\t\t 2. Editing a Particular Book ";
     cout<<"\n\t\t 3. Listing All the Books ";
     cout<<"\n\t\t 4. List of Available Books ";
     cout<<"\n\t\t 5. Sorting Books Based on Title ";
     cout<<"\n\t\t 6. Exit ";
     cout<<"\n\n Enter your Choice ";
     cin>>ch;
     clrscr();
     switch(ch)
     { case 1:
        d.newbook();break;
       case 2:
        d.editbook();break;
       case 3:
        d.listbook();break;
       case 4:
        d.availablebook();break;
       case 5:
        d.sortbook();d.listbook();getch();break;
       case 6:
        exit(0);break;
     }
   }
}
```

**Lab Unit – 9 (2 Hrs Real Time)**

**Ex - 9.1**

```
# include<iostream.h>
class counter
 {
  private :
    unsigned int c;
  public:
    counter()
        {  c = 0; }
    int getcount()
       { return c;}
```

```
    void operator ++()
         { c++;}
    }
void main()
{
    counter c1,c2;
    cout <<  "\n c1 =" << c1.getcount();
    cout <<  "\n c2 =" << c2.getcount();
    c1++;
    c2++;
    ++c2;
    cout << "\n c1 = " << c1.getcount();
    cout << "\n c2 = " << c2.getcount();
}
```

# Ex - 9.2

```
// AREA  & CIRCUMFERENCE OF FOUR FIGS

#include<iostream.h>
#include<conio.h>
#include<math.h>
class shape
  {
   public:
     int l,b1,r,a,b,c;
     double a1;
    void getdata();
    void area1(double);
    void area1(int);
    void area1(int, int);
    void area1(int, int, int);
    };
void shape :: getdata()
{
 cout << " Enter the Values \n";
 cout << "  Side :  ";
 cin >> a1;
 cout << "\n" << " Length :  ";
 cin >> l;
 cout << "\n" << " Breadth :  ";
 cin >> b1;
 cout << "\n" << " Sides of triangle :  ";
 cout << "  a :  ";
 cin >> a;
 cout << "\n\t\t\t" << "b :  ";
 cin >> b;
 cout << "\n\t\t\t" << "c :  ";
 cin >> c;
 cout << "\n" << " Radius :  ";
 cin >> r;
 getch();
 }
 void shape :: area1(double a1)
 {
 cout << "\n\n The Side of Square is \n";
```

```
 cout << " ---------------------- \n";
 cout << "   Side   : \t";
 cout << a1 << "\n";
 cout << " Area of the Square is : \t" << (a1 * a1) << endl;
 cout << " Circumference of the Square is : \t" << (4 *a1) << endl;
 }
 void shape :: area1(int r)
 {
 cout << "\n\n The Radius of the circle is \n";
 cout << " --------------------------- \n";
 cout << "  Radius  : \t";
 cout << r << "\n";
 cout << " Area of the Circle : \t" << (3.14 * r *r) << endl;
 cout << " Circumference of the Circle : \t" << (2 * 3.14 * r) << endl;
 }
void shape :: area1(int l, int b)
{
 cout << "\n\n The Values of the Rectangle \n";
 cout << " --------------------------- \n";
 cout << "  Length : \t";
 cout << l;
 cout << "\n Breadth : \t";
 cout << b;
 cout << "\n Area of the Rectangle : \t" << (l * b) << endl;
 cout << " Circumference of the rectangle : \t" << (2*(l +b)) << endl;
 }
void shape :: area1(int a, int b, int c)
{
 cout << "\n\n The Values of the Triangle \n";
 cout << " -------------------------\n";
 cout << "  A : \t" << a << endl;
 cout << "  B : \t" << b << endl;
 cout << "  C : \t" << c << endl;
 int s,s1;
 s=(a + b + c)/2;
 s1=sqrt(s*(s-a)*(s-b)*(s-c));
 cout << " Area of the Triangle : \t" << s1 << endl;
 cout << " Circumference of the Triangle : \t" << (a + b + c) << endl;
 }
void main()
{
 int r,a,b,l,c,b1;
 double a1;
 clrscr();
 shape t;
 t.getdata();
 while (1)
 {
 clrscr();
 cout << "\n\t\t"<< "     MENU \n";
 cout << "\t\t"<<   "    ------\n";
 cout << "\t\t-----------------\n";
 cout << "\t\t CIRCLE    -1\n";
 cout << "\t\t TRIANGLE  -2\n";
 cout << "\t\t RECTANGLE -3\n";
 cout << "\t\t SQUARE    -4\n";
 cout << "\t\t EXIT      -5\n";
```

```
 cout << "\t\t------------------\n";
 cout << "\n\t Please, Enter Your Choice \t";
 int op;
 cin >> op;
 switch(op)
 {
 case 1:
    t.area1(t.r);
    break;
 case 2:
    t.area1(t.a, t.b, t.c);
    break;
 case 3:
    t.area1(t.l,t.b1);
    break;
 case 4:
    t.area1(t.a1);
    break;
 case 5:
    goto as;
    }
    cout << "\n Press any key to continue \n ";
    getch();
    }
as:;
 }
```

# Ex – 9.3

```
/*   Comparing Two Dates Using Operator Overloading */

# include <iostream.h>
# include <conio.h>
# include <string.h>
# include <stdlib.h>
class date1
{
    char dt[11];
    int dd1,dd2,m1,m2,y1,y2;
    public :
          void getd()
          {
             cout << "\n Enter Date : ";
             cin >> dt;
          }
          friend int operator <= (date1 d1,date1 d2);
          friend void operator -= (date1 d1,date1 d2);
};
int operator<=(date1 d1,date1 d2)
{
   int dd1=atoi(strtok(d1.dt,"-"));
   int m1=atoi(strtok(0,"-"));
   int y1=atoi(strtok(0,""));
   int dd2=atoi(strtok(d2.dt,"-"));
   int m2=atoi(strtok(0,"-"));
   int y2=atoi(strtok(0,""));
   if (y1 == y2)
```

```
    {
      if (m1==m2)
      {
       if (dd1 == dd2)
          return(0);
       else
          if (dd1 > dd2)
             return(1);
          else
             return(-1);
      }
      else
         if (m1 > m2)
            return(1);
         else
            return(-1);
     }
     else
       if (y1 > y2)
          return(1);
       else
          return(-1);
}
void operator -= (date1 d3,date1 d4)
{  int dy, ye, mt;
   int dd1=atoi(strtok(d3.dt,"-"));
   int m1=atoi(strtok(NULL,"-"));
   int y1=atoi(strtok(NULL,""));
   int dd2=atoi(strtok(d4.dt,"-"));
   int m2=atoi(strtok(NULL,"-"));
   int y2=atoi(strtok(NULL,""));
   if((y1==y2) && (m1==m2))
    {if(dd1 > dd2)
      dy = dd1-dd2;
     else
     dy=dd2-dd1;
     mt=0;
     ye=0;
     goto ad;
    }
   else
    {int n1 = (dd1 > dd2)? 0 : 30;
     dy = dd1 - dd2 + n1;}
     if ((y1==y2) && (dd1==dd2))
   {
     if(m1>m2)
      mt=m1-m2;
     else
      mt=m2-m1;
      goto ad;
    }
     else
      { int n2 = ( m1 > m2 ) ? 0 : 12;
       mt = m1 - m2 + n2;}
   if(y1==y2)
   {
    if(m1>m2)
```

```
        { if(dd1>dd2)
          { dy=dd1-dd2;
            mt=m1-m2;
          }
          else
           { dy =(dd1+30)-dd2;
             mt =(m1-1)-m2;
           }
        }
         else
          { if(dd1>dd2)
            { dy =(dd2+30)-dd1;
              mt =(m2-1)-m1;
             }
            else
            { dy=dd2-dd1;
              mt=m2-m1;
             }
          }
            ye=0;
       }
        else
        {  if(y1>y2)
           ye=y1-y2;
           else
           ye=y2-y1;
           }
ad: if(dy>29)
    {mt=mt+1;
     dy=0;
     }
     if(mt>11)
     {ye=ye+1;
      mt=0;
      }
   cout << " \n\n Difference Between Two Dates :\n ";
   cout << "\n Day   : " << dy;
   cout << "\n Month : " << mt;
   cout << "\n Year  : " << ye;
}
void main()
{
    date1 d1,d2;
    int i;
    clrscr();
    d1.getd();
    d2.getd();
    i=d1<=d2;
    if (i==0)
      cout << "\n\n\t\t The Two Dates Are Equal";
    else
    {
      if ( i > 0 )
     {
        cout << "\n\n\t\t The First Date Is Greater";
        d1-=d2;
        }
```

```
      else
      {
        cout << "\n\n\t\t The Second Date Is Greater";
        d2-=d1;
        }
        }
   getch();
}
```

## Ex – 9.4

```
/*             STACK AND QUEUE OPERATIONS
              [using operator overloading]                */

#include <iostream.h>
#include <string.h>
#include <conio.h>
#include <process.h>
class stack
{
  int s[50];
 public:
  int top;
  void operator()(int v);
  int operator()();
  void disp();
};

class queue
{
  int q[50];
 public:
  int front, rear;
  void operator()(int s);
  int operator()();
  void disp();
};

void stack::operator()(int v)
{
 top++;
 s[top]=v;
}

int stack::operator()()
{
 int a = s[top];
 top--;
 return(a);
}

void stack::disp()
{
 cout<<"\t\t    ........\n";
 for(int i=top; i>=1;i--)
   cout<<"\t\t    |  "<<s[i]<<"  |\n";
```

```
 cout<<"\t\t    ........";
}

void queue::operator()(int v1)
{
 rear++;
 q[rear]=v1;
}

int queue::operator()()
{
 int t1=q[front];
 front++;
 return(t1);
}

void queue::disp()
{
 cout<<"\t | ";
 for(int j=front; j<=rear; j++)
   cout <<q[j]<<"  | ";
}

void main()
{
 int t,ch,ch1,ch2;
 char *k;
 stack st;
 queue qu;
 clrscr();
 st.top=0;
 qu.front=1;
 qu.rear=0;
 while(1)
 {getch();
 clrscr();
 cout << "\n\n\t\t\t MENU \n";
 cout << "\t\t\t ****\n\n";
 cout << "\t\t   1. Stack Operations \n";
 cout << "\t\t   2. Queue Operations \n";
 cout << "\t\t   3. Exit \n\n";
 cout << "The Choice : ";
 cin >> ch;
 switch(ch)
 {
 case 1:
   strcpy(k, "y");
   while(strcmp(k, "y")==0)
   {getch();
    clrscr();
   cout << "\n\n\t\t Stack Operations \n";
   cout << "\t\t ~~~~~~~~~~~~~~~~ \n\n";
   cout << "\t\t    1. Push \n";
   cout << "\t\t    2. Pop \n";
   cout << "\t\t    3. Exit \n\n";
   cout << "The choice : ";
   cin >> ch1;
```

```
  switch(ch1)
  {

 case 1:  /* Pushing in the stack */
     cout<<"\n\nEnter the data to be pushed : \n";
     cin >> t;
     st(t);
     cout<<"\n\n\t\tCONTENT OF STACK \n\n";
     st.disp();
     break;
  case 2: /* Popping the elements from stack */
     int e = st();
     cout<<"\n\n\t  After popping an element  "<<e<<"\n";
     cout<<"\n\t       CONTENT OF STACK \n\n";
     st.disp();
     break;
  case 3: strcpy(k, "n");
     break;
    }
  }
  break;
case 2:
    strcpy(k, "y");
    while(strcmp(k, "y")==0)
    { getch();
      clrscr();
      cout << "\n\n\t       Queue Operations\n";
      cout << "\t       ~~~~~~~~~~~~~~~~ \n\n";
      cout << "\t\t1. Insertion \n";
      cout << "\t\t2. Deletion \n";
      cout << "\t\t3. Exit \n\n";
      cout << "The choice : ";
      cin >> ch1;
      switch(ch1)
      {
      case 1:   /* Insertion in the queue */
        cout<<"\n\nEnter the data to be inserted : \n";
        cin >> t;
        qu(t);
        cout<<"\n\n\t\tCONTENT OF QUEUE \n\n";
        qu.disp();
        break;
      case 2: /* Deleting the elements from queue */
        int e = qu();
        cout<<"\n\n\tAfter deleting an element  "<<e<<"\n";
        cout<<"\n\t\tCONTENT OF QUEUE \n\n";
        qu.disp();
        break;
      case 3: strcpy(k, "n");
        break;
    }
  }
  break;
case 3:
     exit(0);
     break;
}
```

```
 }
 getch();
}
```

# Ex – 9.5

```
      /*  String Manipulation using Operator Overloading */

#include <stdio.h>
#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <process.h>
class string1
 {
    char *name;
    int l;
    public :
      string1()
       { l = 0;
        name = new char[1];
       }
      void getstring();
      friend int operator ==(string1,string1);
      int operator ==(string1);
      friend string1 operator +(string1,string1);
      string1 operator +(string1);
      friend string1 operator +(string1,char *);
      string1 operator +(char *);
      friend char * operator +(char *,string1);
      string1 operator =(string1);
      void operator -();
      friend void operator -(string1);
      void putstring(string1 );
  };
void  string1::getstring()
 { char *name1;
   cin>>name1;
   l = strlen(name1);
   name = new char[l+1] ;
   strcpy(name,name1);
   name[l] = '\0';
 }

void  string1::putstring(string1 a)
  { cout<<"\t\t\t"<<a.name;
    cout<<"\n";
  }
int operator ==(string1 a,string1 b)
 {
   if ( strcmp(a.name, b.name) == 0 )
     return 1;
   else
     return 0;
  }
```

```
int  string1::operator ==(string1 a)
 {
   if ( strcmp(name, a.name) == 0 )
     return 1;
   else
     return 0;
  }

string1 operator +(string1 a,string1 b)
 {
   string1 temp;
   temp.l = a.l + b.l;
   temp.name = new char[temp.l + 1];
   strcat(a.name, b.name);
   strcpy(temp.name, a.name);
   return temp;
 }
string1 string1::operator +(string1 a)
 {
   string1 temp;
   temp.l = l + a.l;
   temp.name = new char[temp.l + 1];
   strcat(name, a.name);
   strcpy(temp.name, name);
   return temp;
 }
string1 operator +(string1 a, char *n)
 {  string1 temp;
    temp.l = a.l + strlen(n);
    temp.name = new char[temp.l + 1];
    strcat(a.name, n);
    strcpy(temp.name, a.name);
    return temp;
 }
string1 string1::operator +(char *n)
 {  string1 temp;
    temp.l = l + strlen(n);
    temp.name = new char[temp.l + 1];
    strcat(name, n);
    strcpy(temp.name, name);
    return temp;
 }

char* operator +(char *n,string1 a)
 {
    char *temp;
    strcat(n, a.name);
    strcpy(temp, n);
    temp[strlen(temp)] = '\0';
    return temp;
 }
string1 string1:: operator =(string1 b)
 {
   strcpy(name, b.name);
 }
void string1::operator -()
 {
```

```
    strrev(name);
 }
void operator -(string1 a)
 {
    strrev(a.name);
 }

void main()
 { int t = 0;
  while ( t == 0 )
  {clrscr();
   cout<<"\n\t\t\t Menu \n";
   cout<<"\t\t\t ------\n";
   cout<<"\n\t\t 1.Comparing two Strings ";
   cout<<"\n\t\t 2.Concatenating two String objects ";
   cout<<"\n\t\t 3.Concatenating a string object with given string ";
  cout<<"\n\t\t 4.Concatenating the given string with  a string object ";
   cout<<"\n\t\t 5.Reversing a String ";
   cout<<"\n\t\t 6.Copying a string to another";
   cout<<"\n\t\t 7.Exit ";
   cout<<"\n\n Enter Your Choice ";
   int opt;
   string1 s1,s2,s3;
   char *n,*t1;
   cin>>opt;
   clrscr();
   switch(opt)
   {
   case 1:
     cout<<"\n\n\t\t String  Comparison ";
     cout<<"\n\t\t ------------------ \n";
     cout<<"\n\n Enter two strings to be compared \n";
     cout<<"\nString 1 : ";s1.getstring();
     cout<<"\nString 2 : ";s2.getstring();
     clrscr();
     cout<<"\n\n\t\t String  Comparison ";
     cout<<"\n\t\t ------------------ \n";
     cout<<"\n\n\n\t\t Given Two Strings are : \n\n";
     s1.putstring(s1);
     s2.putstring(s2);
     if ( s1.operator ==(s2) )
       cout<<"\n\n\t\t Two strings are equal \n";
     else
      cout<<"\n\n\t\t  Two strings are not equal \n";
     break;
   case 2:
     cout<<"\n\n\t\t  Concatenating two string Objects ";
     cout<<"\n\t\t ----------------------------------\n\n";
     cout<<"\n\n Enter two strings to be concatenated \n";
     cout<<"String 1 : "; s1.getstring();
     cout<<"\nString 2 : ";s2.getstring();
     clrscr();
     cout<<"\n\n\t\t  Concatenating two string Objects ";
     cout<<"\n\t\t ----------------------------------\n\n";
     cout<<"\n\n\n\t\t Given Two Strings are : \n\n";
     s1.putstring(s1);
     s2.putstring(s2);
```

```
  s3 = operator +(s1,s2);
  cout<<"\n\n\n\t\t String after concatenation is : \n\n";
  s3.putstring(s3);
  break;
case 3:
  cout<<"\n\n\t\t Concatenating a string Object with given string ";
  cout<<"\n\t\t ----------------------------------------";
  cout<<"\n\n Enter two strings to be concatenated \n";
  cout<<"String 1 : ";s1.getstring();
  cout<<"\nString 2 : ";cin>>n;
  clrscr();
  cout<<"\n\n\t\t Concatenating a string Object with given string ";
  cout<<"\n\t\t ----------------------------------------";
  cout<<"\n\n\n\t\t Given Two Strings are : \n\n";
  s1.putstring(s1);
  cout<<"\t\t\t"<<n;
  s3 = s1.operator + (n);
  cout<<"\n\n\n\t\t  String after concatenation is : \n\n";
  s3.putstring(s3);
  break;
case 4:
cout<<"\n\n\t\t Concatenating the given string with a string Objects \n";
  cout<<"\t\t ---------------------------------------- \n\n";
  cout<<"\n\n Enter two strings to be concatenated \n";
  cout<<"\nString 1 : "; cin>>n;
  cout<<"\nString 2 : "; s1.getstring();
  clrscr();
  cout<<"\n\n\t\t Concatenating the given string with a string Objects ";
  cout<<"\n\t\t ----------------------------------------";
  cout<<"\n\n\n\t\t Given Two Strings are : \n\n";
  cout<<"\t\t\t"<<n<<"\n";
  s1.putstring(s1);
  t1 = n + s1;
  cout<<"\n\n\n\t\t String after concatenation is : \n";
  cout<<"\n\t\t\t "<<t1;
  cout<<"\n";
  break;
case 5:
  cout<<"\n\t\t Reversing the given string \n" ;
  cout<<"\t\t---------------------------- \n\n";
  cout<<"\n\t Enter the String to be Reversed ";
  s1.getstring();
  clrscr();
  cout<<"\n\t\t Reversing the given string \n" ;
  cout<<"\t\t---------------------------- \n\n";
  cout<<"\n\n String to be Reversed is  : \n\t";
  s1.putstring(s1);
  s1.operator -();
  cout<<"\n     Reversed string is    : \n\t";
  s1.putstring(s1);
  break;
case 6:
  cout<<"\n\n\t\t Copying a string to another \n";
  cout<<"\t\t --------------------------\n";
  cout<<"\n Enter the string to be copied :  ";
  s1.getstring();
  clrscr();
```

```
     cout<<"\n\n\t\t Copying string A to String B \n";
     cout<<"\t\t ---------------------------- \n\n";
     cout<<"\n String A to be copied   : \n\t";
     s1.putstring(s1);
     s2 = s1;
     cout<<"\n String B after copying  : \n\t";
     s2.putstring(s2);
     break;
   case 7:
     t = 1;
     break;
  }
  getch();
  }
 }
```

# Ex - 9.6

```
#include <iostream.h>
#include<iomanip.h>
#include <conio.h>
class shape
 {  protected :
       double l,b;
    public :
       void getdata(double x,double y );
       double putl(){ return l; }
       double putb(){ return b; }
       virtual void display() {}
  };
void shape::getdata(double x,double y = 0)
 {  l = x;
    b = y;
  }
class triangle : public shape
 {  float area;
    public :
      void display();
  };
void triangle::display()
 {  double l1,b1;
    l1 = putl();
    b1 = putb();
    area = 0.5 * l * b;
    setprecision(2);
    clrscr();
    cout<<"\n\n\n Area of Triangle ";
    cout<<"\n ----------------\n ";
    cout<<"\n\t\t Height    :   "<<setw(5)<<l;
    cout<<"\n\t\t Breadth   :   "<<setw(5)<<b;
    cout<<"\n\t\t Area      :   "<<setw(5)<<area;
 }
class rectangle:public shape
 {  float area;
    public :
       void display();
```

```
  };
void rectangle::display()
 {  clrscr();
    cout<<"\n\n Area of Rectangle";
    cout<<"\n ------------------ \n";
    cout<<"\n\t\t Length   :  "<< l;
    cout<<"\n\t\t Breadth  :  "<<b;
    cout<<"\n\t\t Area     :  "<<l * b;
  }

class circle : public shape
 {  float area;
    public :
      void display();
  };
void circle::display()
 {
    area = 3.414 * l * l;
    clrscr();
    cout<<"\n\n\n Area of Circle ";
    cout<<"\n ---------------\n ";
    cout<<"\n\t\t Radius    :   "<<l;
    cout<<"\n\t\t Area      :   "<<area;
 }
void main()
{  shape *sptr = new shape;
    shape s;
   triangle t;
   rectangle r;
   circle c;
   char t1 = 'y';
   double x,y;
   while ( t1 == 'y' )
    { clrscr();
      cout<<"\n\n\n\t\t\t Menu ";
      cout<<"\n\t\t\t ------- \n";
      cout<<"\n\t\t 1. Area of triangle ";
      cout<<"\n\t\t 2. Area of rectangle";
      cout<<"\n\t\t 3. Area of Circle  ";
      cout<<"\n\t\t 4. Exit";
      cout<<"\n\n\n\t\t Enter your Choice ";
      int opt;
      cin>>opt;
      clrscr();
      switch(opt)

       { case 1:
            cout<<"\n\n\n\t\t Area of triangle ";
            cout<<"\n\t\t-------------------";
            cout<<"\n Enter the height of triangle ";
            cin>>x;
            cout<<"\n Enter the Breadth of triangle ";
            cin>>y;
            sptr = &t;
            sptr->getdata(x,y);
            sptr->display();
            getch();
```

```
            break;

        case 2:
            cout<<"\n\t\t Area of Rectangle ";
            cout<<"\n\t\t ------------------";
            cout<<"\n Enter the length of Rectangle ";
            cin>>x;
            cout<<"\n Enter the Breadth of Rectangle";
            cin>>y;
            sptr = &r;
            sptr->getdata(x,y);
            sptr->display();
            getch();
            break;

        case 3:
            cout<<"\n\t\t Area of Circle ";
            cout<<"\n\t\t --------------";
            cout<<"\n Enter the Radius of Circle ";
            cin>>x;
            sptr = &c;
            sptr->getdata(x,y);
            sptr->display();
            getch();
            break;

         case 4:
            t1 = 'n';
            break;
        }
       }
       }
```

# Ex - 9.7

```
#include<iostream.h>
class basea{
virtual void display(){
cout<<”one”;
}
};
class derivedb:public basea
{
public:
virtual void display()
{
cout<<”two”;
}
}
class derivedc:public derived b
{
public:
virtual void display(){
cout<<”three”
}
};
```

```
void main()
{
basea obja;
derivedb objb;
derivedb objc;
basea *ptr[3];
ptr[0]=obja;
ptr[1]=objb;
ptr[2]=objc;

for (int i =0;i=2;i++)
ptr[i]->display();
}
```

## Ex - 9.8

```
#include<iostream.h>

class base{
       private:
int x;
float y;

       public:
void getdata();
void display();
};

class derivedb:public base
{
        private:
int rollno;
char name[20];

        public:
void getdata();
void display();
};

void base::getdata()
{

           cout<<"Enter the value of x and y";
           cin>>x;
           cin>>y;
}

void base::display()
{
         cout<<"The value of x and y";
         cout<<x;
         cout<<y;
}
void derivedb::getdata()
{
   cout<<"Enter rollno, name";
```

```
        cin>>rollno;
        cin>>name;
}

void derived::display()
{
cout<<rollno<<name;
}

void main()
{
    base *ptr;
   derived b obj;
   ptr = &obj;
   ptr->getdata();
   ptr->display();
}
```

# Ex - 9.9

```
#include<iostream.h>

class base{
        private:
int x;
float y;

        public:
virtual void getdata();
virtual void display();

};

class derivedb:public base
{
         private:
long int rollno;
char name[20];

         public:
void getdata();
void display();
};

void base::getdata()
{

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

}

void base::display()
{
          cout<<"The value of x and y";
```

```
         cout<<x;
         cout<<y;
}

void derivedb::getdata()
{
   cout<<"Enter  rollno, name";
       cin>>rollno;
       cin>>name;
}

void derived::display()
{
cout<<"roll number and name "
cout<<rollno<<name;
}

void main()
{
    base *ptr;
   derivedb obj;
   ptr = &obj;
   ptr->getdata();
   ptr->display();
}
```

## Ex - 9.10

```
#include<iostream.h>

class base
{
       private:
          int x;
          float y;

       public:
     virtual void getdata();
     virtual void display();
   };

class derivedb:public base
{
        private:
             int rollno;
             char name[20];

        public:
             void getdata();
             void display();
};

void base::getdata()
{
  }
```

```
void base::display()
{
}
void derivedb::getdata()
{
      cout<<"Enter  rollno, name ";
       cin>>rollno;
       cin>>name;
}

void derived::display()
{
cout<<rollno<<name;
}

void main()
{
    base *ptr;
   derivedb obj;
   ptr = &obj;
   ptr->getdata();
   ptr->display();
}
```