

A
PRACTICAL RECORD
BOOK
OF
COMPUTER SCIENCE (PCMC)

SECOND PUC



MORARJI DESAI RESIDENTIAL PU
SCIENCE COLLEGE, DUDDA, HASSAN

MORARJI DESAI RESIDENTIAL PU SCIENCE COLLEGE, DUDDA, HASSAN

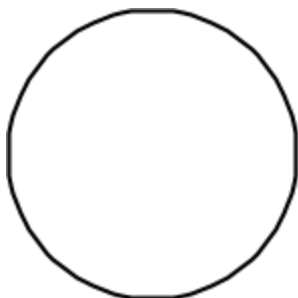


This is to certify that Mr. / Mrs.
has satisfactorily completed the course of experiments in practical
COMPUTER SCIENCE prescribed by the **Pre-University,**
Bangalore for **SECOND PUC (P.C.M.C)** course in the laboratory of
this college in the year 2018-19.

Signature of the Lecturer

Head of the Department

Date:



Name of the Candidate :

Register Number :

Examination Centre :

Date of Practical Examination:

CONTENTS

Program No	Program Name	Page No
SECTION- A (C++ AND DATA STRUCTURES)		
01	Write a C++ program to find the frequency presence of an element in an array.	01
02	Write a C++ program to insert an element into an array at a given position.	03
03	Write a C++ program to delete an element from an array from a given position.	05
04	Write a C++ program to sort the element of an array in ascending order using insertion sort.	07
05	Write a C++ program to search for a given element in an array using binary search method.	09
06	Write a C++ program to create a class with data members principal, time and rate. Create a member function to accept data values, to compute simple interest and to display the result.	11
07	Write a C++ program to create a class with data members a, b, c and member functions to input data, compute the discriminates based on the following conditions and print the roots. ☐ If discriminates = 0, print the roots are equal and their value. ☐ If discriminates > 0, print the real roots and their values. ☐ If discriminates < 0, print the roots are imaginary and exit the program.	13
08	Write a C++ program to find the area of square/ rectangle/ triangle using function overloading.	15
09	Write a C++ program to find cube of a number using inline function.	17
10	Write a C++ program to find sum of the series $1 + x + x^2 + x^3 + \dots x^n$ using constructors.	18
11	Create a base class containing the data member roll number and name. Also create a member function to read and display the data using the concept of single level inheritance. Create a derived class that contains marks of two subjects and total marks as the data members.	20
12	Create a class containing the following data members Register No, Name and Fees. Also create a member function to read and display the data using the concept of pointers to objects.	22
13	Write a C++ program to perform push items into the stack.	24
14	Write a C++ program to perform pop items into the stack.	26
15	Write a C++ program to perform Enqueue and Dequeue.	29

Program No	Program Name	Page No
SECTION – B (SQL)		
01	Generate the electricity bill for one customer Create a table for house hold Electricity bill with the following fields. Insert 10 records into the table. 1. Check the structure of table and note your observation. 2. Add two fields to the table. a. BILL_AMT NUMBER(6,2) b. DUE_DATE DATE 3. Compute the bill amount for each customer as per the following rules. a. MIN_AMT Rs. 50 b. First 100 units Rs 4.50/Unit c. >100 units Rs. 5.50/Unit 4. Compute due date as BILLING_DATE + 15 Days 5. List all the bills generated.	32

02	Create a student database and compute the results. Create a table for class of students with the following fields. 1. Add records into the table for 10 students for Student ID, Student Name and marks in 6 subjects using INSERT command. 2. Display the description of the fields in the table using DESC command. 3. Alter the table and calculate TOTAL and PERC_MARKS. 4. Compute the RESULT as “PASS” or “FAIL” by checking if the student has scored more than 35 marks in each subject. 5. List the contents of the table. 6. Retrieve all the records of the table. 7. Retrieve only ID_NO and S_NAME of all the students. 8. List the students who have result as “PASS”. 9. List the students who have result as “FAIL”. 10. Count the number of students who have passed. 11. Count the number of students who have failed. 12. List the students who have percentage greater than 60. 13. Sort the table according to the order of ID_NO.	35
----	---	----

Field Name	Type
RR_NO	VARCHAR2(10)
CUS_NAME	VARCHAR2(15)
BILLING_DATE	DATE
UNITS	NUMBER(4)

Field Name	Type
ID_NO	NUMBER(4)
S_NAME	VARCHAR2(15)
SUB1	NUMBER(3)
SUB2	NUMBER(3)
SUB3	NUMBER(3)
SUB4	NUMBER(3)
SUB5	NUMBER(3)
SUB6	NUMBER(3)

03	Generate the Employee details and compute the salary based on the department. Create the following table EMPLOYEE Create another table DEPARTMENT Assume the DEPARTMENT names as Purchase (Id-01), Accounts (Id-02), Sales (Id-03), and Apprentice (Id-04) Enter 10 rows of data for table EMPLOYEE and 4 rows of data for DEPARTMENT table. Write the SQL statements for the following: 1. Find the names of all employees who work for the Accounts department. 2. How many employees work for Accounts department? 3. What are the Minimum, Maximum and Average salary of employees working for Accounts department? 4. List the employees working for particular supervisor. 5. Retrieve the department names for each department where only one employee works. 6. Increase the salary of all employees in the sales department by 15%. 7. Add a new Column to the table EMPLOYEE called BONUS NUMBER (5) and compute 5% of the salary to the said field. 8. Delete all the rows for the employee in the Apprentice department.	39
SECTION- C HTML		
01	Write a HTML program to create a study time-table	43
02	Create an HTML program with table and Form.	46

Field Name	Type
EMP_ID	NUMBER(4)
DEPT_ID	NUMBER(2)
EMP_NAME	VARCHAR2(15)
EMP_SALARY	NUMBER(5)

Field Name	Type
DEPT_ID	NUMBER(2)
DEPT_NAME	VARCHAR2(20)
SUPERVISOR	VARCHAR2(20)

SECTION – A

C++ AND DATA STRUCTURES

PROGRAM 1:

Write a C++ program to find the frequency presence of an element in an array.

```
#include<iostream.h>
#include<conio.h>
class Frequency
{
private:
    int a[10], n, ele, count;
public:
    //Data member
    void readdata( );
    void findfreq( );
    //Member functions declaration
    void display( );
};

void Frequency::readdata( )
//Member function definition
{
    cout<<"Enter the size of the array:"<<endl;
    cin>>n;
    cout<<"Enter the array elements:"<<endl;
    for(int i=0; i<n; i++)
        cin>>a[i];
    cout<<"Enter the element to find the frequency"<<endl;
    cin>>ele;
}

void Frequency::findfreq( )
//Member function definition
{
    count=0;
    for(int i=0; i<n; i++)
        if(ele == a[i])
            count++;
//Traversing Operation
}

void Frequency::display( )
//Member function definition
{
    if(count > 0)
        cout<<ele<<" Occurs"<<count<<" Time";
    else
        cout<<ele<<" Does Not Exists";
}

void main( )
//Main function
{
```


Frequency f;

```
clrscr();  
f.readdata();  
f.findfreq();  
f.display();  
getch();  
}
```

OUTPUT 1:

```
Enter the size of the array:  
5  
Enter the array elements  
2      6      2      8      1  
Enter the element to find the frequency:  
2  
2 Occurs 2 Time
```

OUTPUT 2:

```
Enter the size of the array:  
5  
Enter the array elements  
2      7      5      12      9  
Enter the element to find the frequency:  
4  
4 Does Not Exist
```



PROGRAM 2:

Write a C++ program to insert an element into an array at a given position.

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
class Insertion
{
    private:
        int a[10], n, pos, ele, i;
    public:
        //Data Members
        void readdata( );
        void insert( );
        //Member Function Declaration
        void display( );
};

void Insertion::readdata( )
{
    cout<<"Enter the size of the array"<<endl;
    cin>>n;
    cout<<"Enter the elements for the array"<<endl;
    for(i=0; i<n; i++)
        cin>>a[i];
    cout<<"Enter the position of the element in the array"<<endl;
    cin>>pos;
    cout<<"Enter the element to be inserted"<<endl;
    cin>>ele;
}

void Insertion::insert( )
{
    if(pos>n)
    {
        cout
        o
        u
        t
    }
    else
    {
        <
        <
        "
        O
        u
        t
        o
    }
}
```

```
f
a
r
r
a
y
l
i
m
i
t
s
!
!
!
"
;
g
e
t
c
h
(
)
;
exit(0);
```

```
for(i=n; i>=pos; i--)
a[i+1] = a[i];           // Shift array
elements
a[pos] = ele;           // Insert the
given element
n = n+1;                // Size of the
array is incremented by 1
```

```
    }  
}  
  
void Insertion::display( )  
{  
    cout<<"Array elements after insertion are:"<<endl;  
    for(i=0; i<n; i++)  
        cout<<a[i]<<"\t";  
}  
  
void main( )  
{  
    Insertion i;  
    clrscr();  
    i.readdata();  
    i.insert();  
    i.display();  
    getch();  
}
```

OUTPUT 1:

```
Enter the size of the array  
5  
Enter the elements for the array  
5      9      14      16      23  
Enter the position of the element in the array  
4  
Enter the element to be inserted  
35  
Array elements after insertion are:  
5      9      14      16      35      23      _
```

OUTPUT 2:

```
Enter the size of the array  
3  
Enter the elements for the array  
12      8      20  
Enter the position of the element in the array  
4  
Enter the element to be inserted  
15  
Out of array limits!!!
```



PROGRAM 3:

Write a C++ program to delete an element from an array from a given position.

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
class Deletion
{
    private:
        int a[10], n, pos, i;
    public:
        void initialize()
        {
            for(i=0; i<n; i++)
                a[i] = rand()%100;
        }
        void display()
        {
            for(i=0; i<n; i++)
                cout<<a[i]<<" ";
            cout<<endl;
        }
        void deleteElement()
        {
            int i, j;
            if(pos<0 || pos>=n)
            {
                cout<<"Invalid position"<<endl;
                return;
            }
            for(i=pos; i<n; i++)
                a[i] = a[i+1];
            a[i] = 0;
        }
};
```

```

s                                     y
p                                     (
l                                     )
a                                     ;

```

```

void Deletion::readdata( )
{
    cout<<"Enter the size of the array"<<endl;
    cin>>n;
    cout<<"Enter the elements for the array:"<<endl;
        for (i=0; i<n; i++)
            cin>>a[i];
    cout<<"Enter the position to an delete an element:\n";
    cin>>pos;
}

```

```

void Deletion::delet( )
{
    if(pos>n)
    {
        c
        o
        u
        t
    }
    else
    {
        <
        <
        "
        O
        u
        t
        o
        f
        a
        r
        r
        a
        y
        l
        i
        m
        i
        t
        s
        .
        .
        .
    }
}

```

```
!  
!  
!  
"  
;  
g  
e  
t  
c  
h  
(  
)  
;  
exit(0);
```

```
for(i=pos; i<n; i++)  
a[i] = a[i+1];           //  
Move higher position element  
n = n-1;                 //  
Reduce size of the array by 1
```



```
void Deletion::display( )
{
    cout<<"After deletion the array elements are"<<endl;
    for(i=0; i<n; i++)
        cout<<a[i]<<"\t";
}

void main( )
{
    Deletion d;
    clrscr();
    d.readdata( );
    d.delet( );
    d.display( );
    getch( );
}
```

OUTPUT 1:

```
Enter the size of the array
5
Enter the elements for the array:
4      9      14      28      34
Enter the position to an delete an element:
3
After deletion the array elements are
4      9      14      34
```

OUTPUT 2:

```
Enter the size of the array
3
Enter the elements for the array:
9      4      17
Enter the position to an delete an element:
4
Out of array limits...!!!
```



PROGRAM 4:

Write a C++ program to sort the element of an array in ascending order using insertion sort.

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

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

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

```
class Sort
```

```
{
```

```
    private:
```

```
        int a[10], n, i;
```

```
    public:
```

```
};
```

```
v  
o  
i  
d  
r  
e  
a  
d  
d  
a  
t  
a  
(  
)  
;  
v  
o  
i  
d  
i  
n  
s  
e  
r  
t  
i  
o  
n  
s  
o  
r  
t  
(
```

```

)
;
v
o
i
d
d
i
s
p
l
a
y
(
)
;

```

```

void Sort::readdata( )
{
    cout<<"Enter the size of the array:"<<endl;
    cin>>n;
    cout<<"Enter the elements for the array:"<<endl;
    for(i=0; i<n; i++)
        cin>>a[i];
}

```

```

void Sort::insertionsort( )
{
    int j, temp;
    for(i=1; i<n; i++)
    {
        j = i;
        while(j >= 1)
        {
            if(a[j] < a[j-1])
            {
                t
                e
                m
                p
                =
                a
                [
                j
                ]
                ;
                a
                [
                j
                ]
                =
                a
                [
                j
                -
                1
                ]
                ;
                a
                [
                j
                -
                1
                ]
                =
                t
                e
                m
                p
                ;
            }
            j = j-1;
        }
    }
}

```

```
-  
1  
]  
;  
a  
[  
j  
-  
1  
]  
=  
t  
e  
m  
p  
;
```

```

}

void Sort::display( )
{
    for(i=0; i<n; i++)
        cout<<a[i]<<"\t";
    cout<<endl;
}

void main( )
{
    Sort s;
    clrscr();
    s.readdata( );
    cout<<"Unsorted List    "<<endl;
    cout<<"*****"<<endl;
    s.display( );
    s.insertionsort( );
    cout<<"Sorted List      "<<endl;
    cout<<"*****"<<endl;
    s.display( );
    getch( );
}

```

OUTPUT 1:

```

Enter the size of the array:
5
Enter the elements for the array:
25    40    14    8    33
Unsorted List.....
*****
25    40    14    8    33
Sorted List.....
*****
8    14    25    33    40

```

OUTPUT 2:

```

Enter the size of the array:
5
Enter the elements for the array:
-32    -4    0    -5    -23
Unsorted List.....
*****
-32    -4    0    -5    -23
Sorted List.....
*****
-32    -23    -5    -4    0

```



PROGRAM 5:

Write a C++ program to search for a given element in an array using binary search method.

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

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

```
class Search
```

```
{
```

```
    private:
```

```
        int a[10], n, ele, loc, beg, end, mid, i;
```

```
    public:
```

```
        v
```

```
        o
```

```
        i
```

```
        d
```

```
};
```

```
        r
```

```
        e
```

```
        a
```

```
        d
```

```
        d
```

```
        a
```

```
        t
```

```
        a
```

```
        (
```

```
        )
```

```
        ;
```

```
        v
```

```
        o
```

```
        i
```

```
        d
```

```
        b
```

```
        s
```

```
        e
```

```
        a
```

```
        r
```

```
        c
```

```
        h
```

```
        (
```

```
        )
```

```
        ;
```

```
        v
```

```
        o
```

```
        i
```

```
        d
```

```
        d
```

```

i                                     y
s                                     (
p                                     )
l                                     ;
a

```

```

void Search::readdata( )
{
    cout<<"Enter the size of the array:"<<endl;
    cin>>n;
    cout<<"Enter the array elements in sorted order:"<<endl;
        for(i=0;i<n;i++)
            cin>>a[i];
    cout<<"Enter the element to search:"<<endl;
    cin>>ele;
}

```

```

void Search::bsearch( )
{
    loc = -1;                                // Assume that element does not exist
    beg = 0;                                // First element of the array
    end = n-1;                               // Second element of the
    array while(beg <= end)
    {
        mid = (beg+end)/2;
        if(ele == a[mid])                    // Element found at mid
        {
            l
            o
            c
            =
            m
            i
            d
            ;
            b
            r
            e
            a
            k
            ;

            if(ele < a[mid])
                end = mid-1;

```

```
else    beg = mid+1;
```

```
    }  
}
```

```
void Search::display()  
{  
    if(loc == -1)  
        cout<<ele<<" Element does not exist...!!!";  
    else    cout<<ele<<" Found at Location:"<<loc+1;  
}
```

```
void main( )  
{  
    Search s;  
    clrscr( );  
    s.readdata( );  
    s.bsearch( );  
    s.display( );  
    getch( );  
}
```

OUTPUT 1:

```
Enter the size of the array:  
5  
Enter the array elements in sorted order:  
12    23    39    47    57  
Enter the element to search:  
39  
39 Found at Location:3
```

OUTPUT 2:

```
Enter the size of the array:  
4  
Enter the array elements in sorted order:  
5     8     14    17  
Enter the element to search:  
22  
22 Element does not exist...!!!
```



PROGRAM 6:

Write a C++ program to create a class with data members principal, time and rate. Create a member function to accept data values, to compute simple interest and to display the result.

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

class SimpleInterest
{
    private:
        float principal,rate,time,si;
    public:
        void readdata( );
        void compute( );
        void display( );
};

void SimpleInterest::readdata( )
{
    cout<<"Enter the Principal, Rate and Time"<<endl;
    cin>>principal>>rate>>time;
}

void SimpleInterest::compute( )
{
    si=(principal * time * rate)/100;
}

void SimpleInterest::display( )
{
    cout<<"Principal = "<<principal<<endl;
    cout<<"Time = "<<time<<endl;
    cout<<"Rate = "<<rate<<endl;
    cout<<"Simple Interest = "<<si<<endl;
}

void main( )
{
    SimpleInterest
    si; clrscr( );
    si.readdata( );
    si.compute( );
```

```
si.display( );
```

```
    getch();  
}
```

OUTPUT 1:

```
Enter the Principal, Rate and Time  
120000  
12.75  
6.5  
Principal = 120000  
Time = 6.5  
Rate = 12.75  
Simple Interest = 99450
```

OUTPUT 2:

```
Enter the Principal, Rate and Time  
10000  
12  
2  
Principal = 10000  
Time = 2  
Rate = 12  
Simple Interest = 2400
```



PROGRAM 7:

Write a C++ program to create a class with data members a, b, c and member functions to input data, compute the discriminant based on the following conditions and print the roots.

- ☐ **If discriminant = 0, print the roots are equal and their value.**
- ☐ **If discriminant > 0, print the real roots and their values.**
- ☐ **If discriminant < 0, print the roots are imaginary and exit the program.**

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
class Quadratic
{
    private:
        int a, b, c;
        float disc, x, x1, x2;
    public:
        void input();
        void calculateDiscriminant();
        void findRoots();
};
```

```
;  
v  
o  
i  
d  
d  
i  
s  
p  
l  
a  
y  
(  
)  
;
```

```
void Quadratic::readdata( )  
{  
    cout<<"Enter the values for a, b, c (Co-efficeient)"<<endl;  
    cin>>a>>b>>c;  
}
```

```
void Quadratic::compute( )  
{  
    disc = b*b-4*a*c;  
}
```

```
void Quadratic::display( )  
{  
    compute( );  
    if(disc == 0)  
    {  
        cout<<"Equal  
        Roots..."<<endl; x=-b/(2*a);  
        cout<<"Root is.  "<<x;  
    }  
    else if(disc>0)  
    {
```

```
        }
        else
    }

cout<<endl;
//Real and Discriminant Roots...
cout<<endl;
//1.
x1=(-b+sqrt(disc))/(2*a);
x2=(-b-sqrt(disc))/(2*a);
//2.
cout<<endl;
//3.
cout<<endl;
//4.
cout<<endl;
//5.
cout<<endl;
//6.
cout<<endl;
//7.
cout<<endl;
//8.
cout<<endl;
//9.
cout<<endl;
//10.
cout<<endl;
//11.
cout<<endl;
//12.
cout<<endl;
//13.
cout<<endl;
//14.
cout<<endl;
//15.
cout<<endl;
//16.
cout<<endl;
//17.
cout<<endl;
//18.
cout<<endl;
//19.
cout<<endl;
//20.
cout<<endl;
//21.
cout<<endl;
//22.
cout<<endl;
//23.
cout<<endl;
//24.
cout<<endl;
//25.
cout<<endl;
//26.
cout<<endl;
//27.
cout<<endl;
//28.
cout<<endl;
//29.
cout<<endl;
//30.
cout<<endl;
//31.
cout<<endl;
//32.
cout<<endl;
//33.
cout<<endl;
//34.
cout<<endl;
//35.
cout<<endl;
//36.
cout<<endl;
//37.
cout<<endl;
//38.
cout<<endl;
//39.
cout<<endl;
//40.
cout<<endl;
//41.
cout<<endl;
//42.
cout<<endl;
//43.
cout<<endl;
//44.
cout<<endl;
//45.
cout<<endl;
//46.
cout<<endl;
//47.
cout<<endl;
//48.
cout<<endl;
//49.
cout<<endl;
//50.
cout<<endl;
//51.
cout<<endl;
//52.
cout<<endl;
//53.
cout<<endl;
//54.
cout<<endl;
//55.
cout<<endl;
//56.
cout<<endl;
//57.
cout<<endl;
//58.
cout<<endl;
//59.
cout<<endl;
//60.
cout<<endl;
//61.
cout<<endl;
//62.
cout<<endl;
//63.
cout<<endl;
//64.
cout<<endl;
//65.
cout<<endl;
//66.
cout<<endl;
//67.
cout<<endl;
//68.
cout<<endl;
//69.
cout<<endl;
//70.
cout<<endl;
//71.
cout<<endl;
//72.
cout<<endl;
//73.
cout<<endl;
//74.
cout<<endl;
//75.
cout<<endl;
//76.
cout<<endl;
//77.
cout<<endl;
//78.
cout<<endl;
//79.
cout<<endl;
//80.
cout<<endl;
//81.
cout<<endl;
//82.
cout<<endl;
//83.
cout<<endl;
//84.
cout<<endl;
//85.
cout<<endl;
//86.
cout<<endl;
//87.
cout<<endl;
//88.
cout<<endl;
//89.
cout<<endl;
//90.
cout<<endl;
//91.
cout<<endl;
//92.
cout<<endl;
//93.
cout<<endl;
//94.
cout<<endl;
//95.
cout<<endl;
//96.
cout<<endl;
//97.
cout<<endl;
//98.
cout<<endl;
//99.
cout<<endl;
//100.
cout<<endl;
```

```

u          t
t          <
<          <
<          "
"          R
R          o
o          o
o          t
t          2
l          i
i          s
s          "
"          <
<          <
<          x
x          2
l          <
<          <
<          e
e          n
n          d
d          l
l          ;
;          ;
c          cout<<"Imaginary Roots..."<<endl;
o
u

```

```

void main( )
{
    Quadratic
    q; clrscr();
    q.readdata( );
    q.display( );
    getch( );
}

```

OUTPUT 1:

```

Enter the values for a, b, c (Co-efficeient)
2      -7      6
Real and Distinct Roots...
Root 1 is 2
Root 2 is 1.5

```

OUTPUT 2:

```

Enter the values for a, b, c (Co-efficeient)
1      2      1
Equal Roots...
Root is....-1

```

OUTPUT 3:

```
Enter the values for a, b, c (Co-efficeient)
1      2      5
Imaginary Roots...
```



PROGRAM 8:

Write a C++ program to find the area of square/ rectangle/ triangle using function overloading.

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
#include<math.h>

class Funcoverload
{
    public:
        float area(float a)                //To compute area of square
        {
            return a*a;
        }

        float area(float l,float b)        //To compute area of rectangle
        {
            return l*b;
        }

        float area(float s1,float s2,float s3)    //To compute area of triangle
        {
            float s=(s1+s2+s3)/2;
            return sqrt(s*(s-s1)*(s-s2)*(s-s3));
        }
};

void main()
{
    float s1,s2,s3;
    int choice;
    Funcoverload
    f; clrscr( );
    while(1)
    {
        cout<<"Program demonstrates Function Overloaded...!!!"<<endl;
        cout<<"1.To find area of square"<<endl;
        cout<<"2.To find area of rectangle"<<endl;
        cout<<"3.To find the area of triangle"<<endl;
        cout<<"4.Exit"<<endl;
        cout<<"Enter your Choice"<<endl;
        cin>>choice;
```

```

switch(choice)
{
    case 1: cout<<"Enter the input for square"<<endl;
            cin>>s1;
            cout<<"Area of Square= "<<f.area(s1)<<endl;
            break;
    case 2: cout<<"Enter the input for rectangle"<<endl;
            cin>>s1>>s2;
            cout<<"Area of Rectangle= "<<f.area(s1,s2)<<endl;
            break;
    case 3: cout<<"Enter the input for triangle"<<endl;
            cin>>s1>>s2>>s3;
            cout<<"Area of Triangle= "<<f.area(s1,s2,s3)<<endl;
            break;
    case 4: cout<<"End of Program  " <<endl;
            getch();
            exit(1);
    default:cout<<"Invallid Choice  !!!"<<endl;
}
getch( );
}
}

```

OUTPUT:

```

Program demonstrates Function Overloaded...!!!
1.To find area of square
2.To find area of rectangle
3.To find the area of triangle
4.Exit
Enter your Choice
1
Enter the input for square
4
Area of Square= 16
Program demonstrates Function Overloaded...!!!
1.To find area of square
2.To find area of rectangle
3.To find the area of triangle
4.Exit
Enter your Choice
2
Enter the input for rectangle
5      6
Area of Rectangle= 30

```

```

Program demonstrates Function Overloaded...!!!
1.To find area of square
2.To find area of rectangle
3.To find the area of triangle
4.Exit
Enter your Choice
3
Enter the input for triangle
3      4      5
Area of Triangle= 6
Program demonstrates Function Overloaded...!!!
1.To find area of square
2.To find area of rectangle
3.To find the area of triangle
4.Exit
Enter your Choice
4
End of Program.....

```



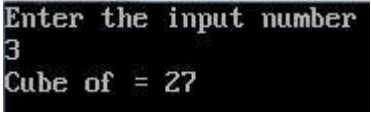
PROGRAM 9:

Write a C++ program to find cube of a number using inline function.

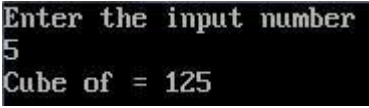
```
#include<iostream.h>
#include<conio.h>

inline int cube(int a)                                //Inline function definition
{
    return a*a*a;
}

void main()                                           //Main Function
{
    int n;
    clrscr( );
    cout<<"Enter the input number"<<endl;
    cin>>n;
    cout<<"Cube of"<<" = "<<cube(n);                //Inline function call
    getch( );
}
```

OUTPUT 1:

```
Enter the input number
3
Cube of = 27
```

OUTPUT 2:

```
Enter the input number
5
Cube of = 125
```



PROGRAM 10:

Write a C++ program to find sum of the series $1 + x + x^2 + x^3 + \dots x^n$ using constructors.

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
class Series
{
    private:
        int sum, x, n;
    public:
        Series(int y, int m)                // Parameterized Constructor
        {
            sum = 1;
            x = y;
            n = m;
        }
        int sumseries( );
};

int Series::sumseries( )
{
    for(int i=1;i<=n;i++)
        sum=sum+pow(x,i);
    return sum;
}

void main()
{
    int x,n;
    clrscr( );
    cout<<"Enter the value for Base(X)="<<endl;
    cin>>x;
    cout<<"Enter the value for Power(n)"<<endl;
```

```
cin>>n;
Series s1(x,n); // Calling Parameterized Constructor
Series s2 = s1; // Copy Constructor
cout<<"Sum of Series using Parameterised Constructor:";
cout<<s1.sumseries()<<endl;
cout<<"Sum of Series using Copy Constructor:";
cout<<s2.sumseries( );
getch( );
}
```

OUTPUT 1:

```
Enter the value for Base(X)=
2
Enter the value for Power(n)
4
Sum of Series using Parameterised Constructor:31
Sum of Series using Copy Constructor:31
```

OUTPUT 2:

```
Enter the value for Base(X)=
3
Enter the value for Power(n)
5
Sum of Series using Parameterised Constructor:364
Sum of Series using Copy Constructor:364_
```



PROGRAM 11:

Create a base class containing the data member roll number and name. Also create a member function to read and display the data using the concept of single level inheritance. Create a derived class that contains marks of two subjects and total marks as the data members.

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

class Student                                     // Base Class
{
    private:
        long
        rollnumber; char
        name[20];
    public:
        void readdata( )
        {
            cout<<"Enter the Roll Number: ";
            cin>>rollnumber;
            cout<<"Enter the Student Name:";
            cin>>name;
        }
        void display( )
        {
            cout<<"\nRoll Number
            :"<<rollnumber<<endl; cout<<"Student
            Name:"<<name<<endl;
        }
};

class Report : public Student                     // Derived class
{
    private:
        int marks1, marks2, total;
    public:
        void readmarks( )
        {
            cout<<"\nEnter Subject 1 Marks: ";
            cin>>marks1;
            cout<<"Enter Subject 2 Marks: ";
            cin>>marks2;
        }
        void compute( )
        {
            total = marks1 + marks2;
```

```
        cout<<endl<<"Total Marks : "<<total;
    }
};

void main( )
{
    Report R;                // Create an object R to process Student
    data clrscr( );
    R.readdata( );
    R.display( );
    R.readmarks(
    ); R.compute(
    );
    getch( );
}
```

OUTPUT 1:

```
Enter the Roll Number: 243850
Enter the Student Name:Keerthi

Roll Number      :243850
Student Name     :Keerthi

Enter Subject 1 Marks: 89
Enter Subject 2 Marks: 92

Total Marks   : 181_
```

OUTPUT 2:

```
Enter the Roll Number: 123456
Enter the Student Name:Akash

Roll Number      :123456
Student Name     :Akash

Enter Subject 1 Marks: 65
Enter Subject 2 Marks: 78

Total Marks   : 143
```



PROGRAM 12:

Create a class containing the following data members Register_No, Name and Fees. Also create a member function to read and display the data using the concept of pointers to objects.

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

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

```
class Student
```

```
{
```

```
    private:
```

```
    public:
```

```
};
```

```
l  
o  
n  
g  
r  
e  
g  
i  
s  
t  
e  
r  
_  
N  
o  
;  
c  
h  
a  
r  
a  
c  
t  
e  
r  
s  
;  
v  
o  
i  
d
```



```
readdata()
{
    int regno;
    string name;
    int fees;
}

display()
{
    cout<<"Register Number : ";
    cout<<regno<<endl;
    cout<<"Student Name : ";
    cout<<name<<endl;
    cout<<"Fees : ";
    cout<<fees<<endl;
}

main()
{
    Student *S; // Create a pointer to point Student object
    clrscr();
    S->readdata(); // Access Student data member using a pointer
    S->display(); // Display data using a pointer
    getch();
}
```

OUTPUT 1:

```
Enter the Register Number:
243850
Enter the Student Name:
Keerthi
Enter the Fees:
14050
Register Number : 243850
Student Name    : Keerthi
Fees            : 14050
```

OUTPUT 2:

```
Enter the Register Number:
12345
Enter the Student Name:
Akash
Enter the Fees:
25000
Register Number : 12345
Student Name    : Akash
Fees            : 25000
```



PROGRAM 13:

Write a C++ program to perform push items into the stack.

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
#define MAX 3
class Stack
{
    private:
    int s[MAX], top;
    public:
        Stack( )                // Constructor to initialize TOP pointer
        {
            top = -1;
        }
        void push(int);          // Member Function Declaration
        void display( );
};
void Stack::push(int item)
{
    if(top == MAX-1)
        cout<<"Stack is Full   Overflow!!!"<<endl;
    else
    {
        t
        o
        p
        +
        +
        ;
        s
        [
        t
        o
        p
        ]
        =
        i
        t
        e
        m
        ;

        void Stack::display( )
        {
            if(top == -1)
```

```
        cout<<"Empty Stack!!!"<<endl;
    else
    {
        for(int i=0; i<=top; i++)
            cout<<endl<<s[i];
        cout<<"-->top element"<<endl;
    }

    getch( );
}
void main( )
{
    Stack s;
    int choice, ele;
```

```

clrscr(
);
while(1)
{
    cout<<"Stack Push Operation Menu"<<endl;
    cout<<"1. PUSH"<<endl;
    cout<<"2. DISPLAY"<<endl;
    cout<<"3. EXIT"<<endl;
    cout<<"Enter your Choice"<<endl;
    cin>>choice;
    switch(choice)
    {
        case 1: cout<<"Push Operation"<<endl;
                cout<<"Enter the value of element:"<<endl;
                cin>>ele;
                s.push(ele);
                break;
        case 2: cout<<"Stack elements are:"<<endl;
                s.display( );
                break;
        case 3: cout<<"End of Stack Operation"<<endl;
                getch( );
                exit(1);
        default:cout<<"Invalid choice   !!!"<<endl;
    }
    getch();
}
}

```

OUTPUT:

```

Stack Push Operation Menu
1. PUSH
2. DISPLAY
3. EXIT
Enter your Choice
2
Stack elements are:
Empty Stack!!!
Stack Push Operation Menu
1. PUSH
2. DISPLAY
3. EXIT
Enter your Choice
1
Push Operation
Enter the value of element:
10
Stack Push Operation Menu
1. PUSH
2. DISPLAY
3. EXIT
Enter your Choice
1

```

```

Push Operation
Enter the value of element:
20
Stack Push Operation Menu
1. PUSH
2. DISPLAY
3. EXIT
Enter your Choice
1
Push Operation
Enter the value of element:
30
Stack Push Operation Menu
1. PUSH
2. DISPLAY
3. EXIT
Enter your Choice
1
Push Operation
Enter the value of element:
40
Stack is Full....Overflow!!!

```

```

Stack Push Operation Menu
1. PUSH
2. DISPLAY
3. EXIT
Enter your Choice
2
Stack elements are:
10
20
30-->top element
Stack Push Operation Menu
1. PUSH
2. DISPLAY
3. EXIT
Enter your Choice
3
End of Stack Operation

```

PROGRAM 14:

Write a C++ program to perform pop items into the stack.

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
#define MAX 3
class Stack
{
    private:
        int s[MAX], top;
    public:
        Stack()                // Constructor to initialize TOP pointer
        {
            top = -1;
        }
        void push(int);
        int pop( );            // Member Functions
        Declaration void display( );
};

void Stack::push(int item)
{
    if(top == MAX-1)
        cout<<"Stack is Full  Overflow!!!"<<endl;
    else
    {
        t
        o
        p
        +
        +
        ;
        s
        [
        t
        o
        p
        ]
        =
        i
        t
        e
        m
        ;
    }
}
```

```
int Stack::pop()
{
    int item;
    if(top == -1)
        cout<<"Stack Empty!!!...Can't POP"<<endl;
    else
    {
        i
        t
        e
        m
        =
        s
        [
        t
        o
        p
        ]
        ;
        t
        o
        p
        -
        -
        ;

        return item;
    }
}
```

```
void Stack::display( )
{
    if(top == -1)
        cout<<"Stack Empty!!!"<<endl;
    else
    {
        for
        (
            i
            =
            0
            ;
            i
            <
            =
            t
            o
            p
            ;
            i
            +
            +
            )
            c
            o
            u
            t
            <
            <
            e
            n
            d
            l
            <
            <
            s
            [
            i
            ]
            ;
        cout<<"-->top element"<<endl;
    }
}
```

```
void main( )
```




```
{
    Stack s;
    int choice, ele;
    clrscr();
    while(1)
    {
        cout<<"\n Stack Push & Pop Operation Menu"<<endl;
        cout<<"1.PUSH"<<endl;
        cout<<"2.POP"<<endl;
        cout<<"3.DISPLAY"<<endl;
        cout<<"4.EXIT"<<endl;
        cout<<"Enter your Choice"<<endl;
        cin>>choice;
        switch(choice)
        {
            case 1: cout<<"Push Operation"<<endl;
                    cout<<"enter the value of an  
                    element"<<endl; cin>>ele;
                    s.push(ele);
                    break;
            case 2: cout<<"Pop Operation"<<endl;
                    cout<<"Popped Element is: "<<s.pop( );
                    break;
            case 3: cout<<"Stack elements are:"<<endl;
                    s.display( );
                    break;
            case 4: cout<<"End of the Stack Operetion"<<endl;
                    getch( );
                    exit(1);
            default:cout<<"Invalid  
                    Choice...!!!"<<endl; break;
        }
        getch();
    }
}
```

}

OUTPUT:

Stack Push & Pop Operation Menu 1.PUSH 2.POP 3.DISPLAY 4.EXIT Enter your Choice 2 Pop Operation Popped Element is: Stack Empty!!!...Can't POP	Stack Push & Pop Operation Menu 1.PUSH 2.POP 3.DISPLAY 4.EXIT Enter your Choice 1 Push Operation enter the value of an element 10	Stack Push & Pop Operation Menu 1.PUSH 2.POP 3.DISPLAY 4.EXIT Enter your Choice 1 Push Operation enter the value of an element 30
Stack Push & Pop Operation Menu 1.PUSH 2.POP 3.DISPLAY 4.EXIT Enter your Choice 3 Stack elements are: Stack Empty!!!	Stack Push & Pop Operation Menu 1.PUSH 2.POP 3.DISPLAY 4.EXIT Enter your Choice 1 Push Operation enter the value of an element 20	Stack Push & Pop Operation Menu 1.PUSH 2.POP 3.DISPLAY 4.EXIT Enter your Choice 1 Push Operation enter the value of an element 40 Stack is Full....Overflow!!!

Stack Push & Pop Operation Menu 1.PUSH 2.POP 3.DISPLAY 4.EXIT Enter your Choice 3 Stack elements are: 10 20 30-->top element	Stack Push & Pop Operation Menu 1.PUSH 2.POP 3.DISPLAY 4.EXIT Enter your Choice 3 Stack elements are: 10 20-->top element
Stack Push & Pop Operation Menu 1.PUSH 2.POP 3.DISPLAY 4.EXIT Enter your Choice 2 Pop Operation Popped Element is: 30_	Stack Push & Pop Operation Menu 1.PUSH 2.POP 3.DISPLAY 4.EXIT Enter your Choice 4 End of the Stack Operation



PROGRAM 15:**Write a C++ program to perform Enqueue and Dequeue.**

```

#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
#define MAX 3
class Queue
{
    private:
        int q[MAX],front,rear;
    public:
        Queue()                // Constructor to initialize FRONT and REAR pointer
        {
            front = -1;
            rear = -1;
        }
        void enqueue(int);
        int dequeue();    // Member Functions
        Declaration void display();
};
void Queue::enqueue(int item)
{
    if(rear == MAX-1)
    {
        cout<<"Queue is full      Overflow!!!"<<endl;
        getch( );
        exit(0);
    }
    if(front == -1)
    {
        front = 0;
        rear = 0;
    }
    else
    {
        r
        e
        a
        r
        +
        +
        ;
        q
        [
        r

```

```
e      t
a      e
r      m
]      ;
=      cout<<"Item Inserted: "<<item<<endl;
i
```

```
int Queue::dequeue()
{
    int item;
    if(front == -1)
    {
```

```
        cout<<"Queue is Empty Underflow!!!"<<endl;
        //getch();
        //exit(1);
    }
    item = q[front];
    if(front ==
    rear)
    {
        front = -1;
        rear = -1;
    }
    else
        front++;

    return item;
}

void Queue::display()
{
    if(front == -1)
        cout<<"Queue is Empty!!!"<<endl;
    else
        for(int i=front; i<=rear; i++)
            cout<<q[i]<<endl;
}

void main()
{
    int ele, choice;
    Queue q;
    clrscr( );
    while(1)
    {
        cout<<"\nQueue Operation Menu"<<endl;
        cout<<"1.Adding Element"<<endl;
        cout<<"2.Deleting Element"<<endl;
        cout<<"3.Display"<<endl;
        cout<<"4.Exit"<<endl;
        cout<<"Enter your Choice"<<endl;
        cin>>choice;
        switch(choice)
        {
            case 1: cout<<"Enter the element to be inserted"<<endl;
                    cin>>ele;
                    q.enqueue(ele);
                    break;
            case 2: cout<<"Deleted Item = "<<q.dequeue( );
```



break;

```

        case 3: cout<<"The Queue Contents:"<<endl;
                q.display( );
                break;
        case 4: cout<<"End of Queue Operation"<<endl;
                getch();
                exit(1);
        default:cout<<"Invalid Choice...!!!"<<endl;
    }
    getch();
}
}

```

OUTPUT:

```

Queue Operation Menu
1.Adding Element
2.Deleting Element
3.Display
4.Exit
Enter your Choice
2
Deleted Item =
Queue is Empty...Underflow!!!

Queue Operation Menu
1.Adding Element
2.Deleting Element
3.Display
4.Exit
Enter your Choice
3
The Queue Contents:
Queue is Empty!!!

```

```

Queue Operation Menu
1.Adding Element
2.Deleting Element
3.Display
4.Exit
Enter your Choice
1
Enter the element to be inserted
10
Item Inserted: 10

Queue Operation Menu
1.Adding Element
2.Deleting Element
3.Display
4.Exit
Enter your Choice
1
Enter the element to be inserted
20
Item Inserted: 20

```

```

Queue Operation Menu
1.Adding Element
2.Deleting Element
3.Display
4.Exit
Enter your Choice
1
Enter the element to be inserted
30
Item Inserted: 30

Queue Operation Menu
1.Adding Element
2.Deleting Element
3.Display
4.Exit
Enter your Choice
1
Enter the element to be inserted
40
Queue is full.....Overflow!!!

```

```

Queue Operation Menu
1.Adding Element
2.Deleting Element
3.Display
4.Exit
Enter your Choice
3
The Queue Contents:
10
20
30

```

```

Queue Operation Menu
1.Adding Element
2.Deleting Element
3.Display
4.Exit
Enter your Choice
2
Deleted Item = 10

```

```

Queue Operation Menu
1.Adding Element
2.Deleting Element
3.Display
4.Exit
Enter your Choice
2
Deleted Item = 20
Queue Operation Menu
1.Adding Element
2.Deleting Element
3.Display
4.Exit
Enter your Choice
3
The Queue Contents:
30

```



SECTION – B

***STRUCTURED QUERY LANGUAGE
(SQL)***

PROGRAM 1:

Generate the electricity bill for one customer.

Create a table for house hold Electricity bill with the following fields.

Field Name	Type
RR_NO	VARCHAR2(10)
CUS_NAME	VARCHAR2(15)
BILLING_DATE	DATE
UNITS	NUMBER(4)

Insert 10 records into the table.

1. Check the structure of table and note your observation.
2. Add two fields to the table.
 - a. BILL_AMT NUMBER(6,2)
 - b. DUE_DATE DATE
3. Compute the bill amount for each customer as per the following rules.
 - a. MIN_AMT Rs. 50
 - b. First 100 units Rs 4.50/Unit
 - c. >100 units Rs. 5.50/Unit
4. Compute due date as BILLING_DATE + 15 Days
5. List all the bills generated.

Solution:

First we have to create the table **EBILL** using **CREATE TABLE** command.

```
SQL> CREATE TABLE EBILL
2  (
3    RR_NO  VARCHAR2(10),
4    CUS_NAME VARCHAR(15),
5    BILLING_DATE DATE,
6    UNITS  NUMBER(4)
7  );
```

Table created.

Insert 10 records into the table using INSERT commands

```
SQL> INSERT INTO EBILL VALUES ('EH 1003', 'ARUN KUMAR', '12-MAR-16', 98);
SQL> INSERT INTO EBILL VALUES ('EH 2005', 'NAVEEN', '14-MAR-16', 108);
SQL> INSERT INTO EBILL VALUES ('EH 2007', 'VARUN', '18-FEB-16', 157);
SQL> INSERT INTO EBILL VALUES ('EH 3009', 'DAVID', '11-APR-16', 77);
SQL> INSERT INTO EBILL VALUES ('EH 3010', 'JHON', '01-MAR-16', 89);
SQL> INSERT INTO EBILL VALUES ('EH 3013', 'AKSHAY', '02-FEB-16', 68);
SQL> INSERT INTO EBILL VALUES ('EH 1010', 'CHANDRU', '12-MAR-16', 108);
SQL> INSERT INTO EBILL VALUES ('EH 1008', 'GHANAVI', '12-MAR-16', 132);
SQL> INSERT INTO EBILL VALUES ('EH 2105', 'DRUVA', '12-MAR-16', 87);
```

```
SQL> INSERT INTO EBILL VALUES ('EH 3041', 'SHREYA', '12-MAR-16', 127);
```

```
SQL> SELECT * FROM EBILL;
```

RR_NO	CUS_NAME	BILLING_D	UNITS
EH 1003	ARUN KUMAR	12-MAR-16	98
EH 2005	NAVEEN	14-MAR-16	108
EH 2007	VARUN	18-FEB-16	157
EH 3009	DAVID	11-APR-16	77
EH 3010	JHON	01-MAR-16	89
EH 3013	AKSHAY	02-FEB-16	68
EH 1010	CHANDRU	12-MAR-16	108
EH 1008	GHANAVI	12-MAR-16	132
EH 2105	DRUVA	12-MAR-16	87
EH 3041	SHREYA	12-MAR-16	127

10 rows selected.

1. Check the structure of table and note your observation.

```
SQL> DESC EBILL;
```

Name	Null?	Type
RR_NO		VARCHAR2(10)
CUS_NAME		VARCHAR2(15)
BILLING_DATE		DATE
UNITS		NUMBER(4)

2. Add two fields to the table.

- BILL_AMT NUMBER(6,2)
- DUE_DATE DATE

```
SQL> ALTER TABLE EBILL ADD(BILL_AMT NUMBER(6,2));
```

Table altered.

```
SQL> ALTER TABLE EBILL ADD(DUE_DATE DATE);
```

Table altered.

3. Compute the bill amount for each customer as per the following rules.

- MIN_AMT Rs. 50
- First 100 units Rs 4.50/Unit
- >100 units Rs. 5.50/Unit

COMMAND 1:

```
SQL> UPDATE EBILL SET BILL_AMT=100 + UNITS *4.25 WHERE UNITS <=100;
```

5 rows updated.

COMMAND 2:

```
SQL> UPDATE EBILL SET BILL_AMT=100 + 100 *4.25 + (UNITS -100) *5 WHERE UNITS >100;
```

5 rows updated.

4. Compute due date as BILLING_DATE + 15 Days

```
SQL> UPDATE EBILL SET DUE_DATE = BILLING_DATE + 15;
```

10 rows updated.

5. List all the bills generated.

```
SQL> SELECT * FROM EBILL;
```

RR_NO	CUS_NAME	BILLING_D	UNITS	BILL_AMT	DUE_DATE
EH 1003	ARUN KUMAR	12-MAR-16	98	516.5	27-MAR-16
EH 2005	NAVEEN	14-MAR-16	108	565	29-MAR-16
EH 2007	VARUN	18-FEB-16	157	810	04-MAR-16
EH 3009	DAVID	11-APR-16	77	427.25	26-APR-16
EH 3010	JHON	01-MAR-16	89	478.25	16-MAR-16
EH 3013	AKSHAY	02-FEB-16	68	389	17-FEB-16
EH 1010	CHANDRU	12-MAR-16	108	565	27-MAR-16
EH 1008	GHANAVI	12-MAR-16	132	685	27-MAR-16
EH 2105	DRUVA	12-MAR-16	87	469.75	27-MAR-16
EH 3041	SHREYA	12-MAR-16	127	660	27-MAR-16

10 rows selected.



PROGRAM 2:

Create a student database and compute the results.

Create a table for class of students with the following fields.

Field Name	Type
ID_NO	NUMBER(4)
S_NAME	VARCHAR2(15)
SUB1	NUMBER(3)
SUB2	NUMBER(3)
SUB3	NUMBER(3)
SUB4	NUMBER(3)
SUB5	NUMBER(3)
SUB6	NUMBER(3)

1. Add records into the table for 10 students for Student ID, Student Name and marks in 6 subjects using INSERT command.
2. Display the description of the fields in the table using DESC command.
3. Alter the table and calculate TOTAL and PERC_MARKS.
4. Compute the RESULT as "PASS" or "FAIL" by checking if the student has scored more than 35 marks in each subject.
5. List the contents of the table.
6. Retrieve all the records of the table.
7. Retrieve only ID_NO and S_NAME of all the students.
8. List the students who have result as "PASS".
9. List the students who have result as "FAIL".
10. Count the number of students who have passed.
11. Count the number of students who have failed.
12. List the students who have percentage greater than 60.
13. Sort the table according to the order of ID_NO.

Solution:

First we have to create the table **CLASS** using **CREATE TABLE** command.

```
SQL> CREATE TABLE CLASS
2  (
3    ID_NO    NUMBER(4),
4    S_NAME   VARCHAR2(15),
5    SUB1     NUMBER(3),
6    SUB2     NUMBER(3),
7    SUB3     NUMBER(3),
8    SUB4     NUMBER(3),
9    SUB5     NUMBER(3),
10   SUB6     NUMBER(3)
11  );
```

Table created.

1. Add records into the table for 10 students for Student ID, Student Name and marks in 6 subjects using INSERT command.

```
SQL> INSERT INTO CLASS VALUES (1401, 'PAWAN', 56, 36, 56, 78, 44, 67);
```

```
SQL> INSERT INTO CLASS VALUES (1411, 'RAJESH', 100, 100, 96, 100, 100, 100);
```

```
SQL>INSERT INTO CLASS VALUES (1412, 'KARAN', 60,30,45,45,36,49);
SQL>INSERT INTO CLASS VALUES (1403, 'SACHIN', 56,60,72,57,78,67);
SQL>INSERT INTO CLASS VALUES (1410, 'PRAKASH', 96,99,97,90,78,100);
SQL>INSERT INTO CLASS VALUES (1402, 'POOJA', 30,45,39,20,33,56);
SQL>INSERT INTO CLASS VALUES (1405, 'ASHWINI', 79,65,79,70,89,88);
SQL>INSERT INTO CLASS VALUES (1406, 'PRAJWAL', 100,90,100,89,90,100);
SQL>INSERT INTO CLASS VALUES (1404, 'BALU', 35,30,78,23,44,70);
SQL>INSERT INTO CLASS VALUES (1407, 'ESHWAR', 100,100,100,98,99,100);
```

2. Display the description of the fields in the table using DESC command.

```
SQL> DESC CLASS;
```

Name	Null?	Type
ID_NO		NUMBER(4)
S_NAME		VARCHAR2(15)
SUB1		NUMBER(3)
SUB2		NUMBER(3)
SUB3		NUMBER(3)
SUB4		NUMBER(3)
SUB5		NUMBER(3)
SUB6		NUMBER(3)

3. Alter the table and calculate TOTAL and PERC_MARKS.

```
SQL> ALTER TABLE CLASS ADD
2 (TOTAL NUMBER(3), PERC_MARKS NUMBER(6,2), RESULT VARCHAR2(10));
```

Table altered.

```
SQL> UPDATE CLASS SET TOTAL = SUB1+SUB2+SUB3+SUB4+SUB5+SUB6;
```

10 rows updated.

```
SQL> UPDATE CLASS SET PERC_MARKS = TOTAL/6;
```

10 rows updated.

4. Compute the RESULT as “PASS” or “FAIL” by checking if the student has scored more than 35 marks in each subject.

```
SQL> UPDATE CLASS SET RESULT = 'PASS'
2 WHERE (SUB1>=35 AND SUB2>=35 AND SUB3>=35 AND SUB4>=35 AND SUB5>=35 AND SUB6>=35);
```

7 rows updated.

```
SQL>
```

```
SQL> UPDATE CLASS SET RESULT = 'FAIL'
2 WHERE (SUB1<35 OR SUB2<35 OR SUB3<35 OR SUB4<35 OR SUB5<35 OR SUB6<35);
```

3 rows updated.

5. List the contents of the table.

6. Retrieve all the records of the table.

```
SQL> SELECT * FROM CLASS;
```

ID_NO	S_NAME	SUB1	SUB2	SUB3	SUB4	SUB5	SUB6	TOTAL	PERC_MARKS	RESULT
1401	PAWAN	56	36	56	78	44	67	337	56.17	PASS
1411	RAJESH	100	100	96	100	100	100	596	99.33	PASS
1412	KARAN	60	30	45	45	36	49	265	44.17	FAIL
1403	SACHIN	56	60	72	57	78	67	390	65	PASS
1410	PRAKASH	96	99	97	90	78	100	560	93.33	PASS
1402	POOJA	30	45	39	20	33	56	223	37.17	FAIL
1405	ASHWINI	79	65	79	70	89	88	470	78.33	PASS
1406	PRAJWAL	100	90	100	89	90	100	569	94.83	PASS
1404	BALU	35	30	78	23	44	70	280	46.67	FAIL
1407	ESHWAR	100	100	100	98	99	100	597	99.5	PASS

10 rows selected.

7. Retrieve only ID_NO and S_NAME of all the students.

```
SQL> SELECT ID_NO, S_NAME FROM CLASS;
```

ID_NO	S_NAME
1401	PAWAN
1411	RAJESH
1412	KARAN
1403	SACHIN
1410	PRAKASH
1402	POOJA
1405	ASHWINI
1406	PRAJWAL
1404	BALU
1407	ESHWAR

10 rows selected.

8. List the students who have result as “PASS”.

```
SQL> SELECT * FROM CLASS WHERE RESULT='PASS';
```

ID_NO	S_NAME	SUB1	SUB2	SUB3	SUB4	SUB5	SUB6	TOTAL	PERC_MARKS	RESULT
1401	PAWAN	56	36	56	78	44	67	337	56.17	PASS
1411	RAJESH	100	100	96	100	100	100	596	99.33	PASS
1403	SACHIN	56	60	72	57	78	67	390	65	PASS
1410	PRAKASH	96	99	97	90	78	100	560	93.33	PASS
1405	ASHWINI	79	65	79	70	89	88	470	78.33	PASS
1406	PRAJWAL	100	90	100	89	90	100	569	94.83	PASS
1407	ESHWAR	100	100	100	98	99	100	597	99.5	PASS

7 rows selected.

9. List the students who have result as “FAIL”.

```
SQL> SELECT * FROM CLASS WHERE RESULT='FAIL';
```

ID_NO	S_NAME	SUB1	SUB2	SUB3	SUB4	SUB5	SUB6	TOTAL	PERC_MARKS	RESULT
1412	KARAN	60	30	45	45	36	49	265	44.17	FAIL
1402	POOJA	30	45	39	20	33	56	223	37.17	FAIL
1404	BALU	35	30	78	23	44	70	280	46.67	FAIL

10. Count the number of students who have passed.

```
SQL> SELECT COUNT(*) FROM CLASS WHERE RESULT = 'PASS';
```

COUNT(*)
7

11. Count the number of students who have failed.

```
SQL> SELECT COUNT(*) FROM CLASS WHERE RESULT='FAIL';
```

```

COUNT(*)
-----
          3

```

12. List the students who have percentage greater than 60.

```
SQL> SELECT * FROM CLASS WHERE PERC_MARKS>60;
```

ID_NO	S_NAME	SUB1	SUB2	SUB3	SUB4	SUB5	SUB6	TOTAL	PERC_MARKS	RESULT
1411	RAJESH	100	100	96	100	100	100	596	99.33	PASS
1403	SACHIN	56	60	72	57	78	67	390	65	PASS
1410	PRAKASH	96	99	97	90	78	100	560	93.33	PASS
1405	ASHWINI	79	65	79	70	89	88	470	78.33	PASS
1406	PRAJWAL	100	90	100	89	90	100	569	94.83	PASS
1407	ESHWAR	100	100	100	98	99	100	597	99.5	PASS

6 rows selected.

13. Sort the table according to the order of ID_NO.

```
SQL> SELECT * FROM CLASS ORDER BY ID_NO;
```

ID_NO	S_NAME	SUB1	SUB2	SUB3	SUB4	SUB5	SUB6	TOTAL	PERC_MARKS	RESULT
1401	PAVAN	56	36	56	78	44	67	337	56.17	PASS
1402	POOJA	30	45	39	20	33	56	223	37.17	FAIL
1403	SACHIN	56	60	72	57	78	67	390	65	PASS
1404	BALU	35	30	78	23	44	70	280	46.67	FAIL
1405	ASHWINI	79	65	79	70	89	88	470	78.33	PASS
1406	PRAJWAL	100	90	100	89	90	100	569	94.83	PASS
1407	ESHWAR	100	100	100	98	99	100	597	99.5	PASS
1410	PRAKASH	96	99	97	90	78	100	560	93.33	PASS
1411	RAJESH	100	100	96	100	100	100	596	99.33	PASS
1412	KARAN	60	30	45	45	36	49	265	44.17	FAIL

10 rows selected.



PROGRAM 3:

Generate the Employee details and compute the salary based on the department.

Create the following table EMPLOYEE.

Field Name	Type
EMP_ID	NUMBER(4)
DEPT_ID	NUMBER(2)
EMP_NAME	VARCHAR2(10)
EMP_SALARY	NUMBER(5)

Create another table DEPARTMENT.

Field Name	Type
DEPT_ID	NUMBER(2)
DEPT_NAME	VARCHAR2(10)
SUPERVISOR	VARCHAR2(10)

Assume the DEPARTMENT names as Purchase (Id-01), Accounts (Id-02), Sales (Id-03), and Apprentice (Id-04)

Enter 10 rows of data for table EMPLOYEE and 4 rows of data for DEPARTMENT table.

Write the SQL statements for the following:

1. Find the names of all employees who work for the Accounts department.
2. How many employees work for Accounts department?
3. What are the Minimum, Maximum and Average salary of employees working for Accounts department?
4. List the employees working for particular supervisor.
5. Retrieve the department names for each department where only one employee works.
6. Increase the salary of all employees in the sales department by 15%.
7. Add a new Column to the table EMPLOYEE called BONUS NUMBER (5) and compute 5% of the salary to the said field.
8. Delete all the rows for the employee in the Apprentice department.

Solution:

First we have to create two tables, **EMPLOYEE** and **DEAPRTMENT**.

<pre>SQL> CREATE TABLE EMPLOYEE 2 (3 EMP_ID NUMBER(4), 4 DEPT_ID NUMBER(2), 5 EMP_NAME VARCHAR2(10), 6 EMP_SALARY NUMBER(5) 7);</pre> Table created.	<pre>SQL> CREATE TABLE DEPARTMENT 2 (3 DEPT_ID NUMBER(2), 4 DEPT_NAME VARCHAR2(10), 5 SUPERVISOR VARCHAR2(10) 6);</pre> Table created.
---	---

To Insert 10 records into the table EMPLOYEE using INSERT INTO command.

```
SQL> INSERT INTO EMPLOYEE VALUES (101, 01, 'ARUN', 15000);
```

```
SQL> INSERT INTO EMPLOYEE VALUES (104, 02, 'MOHAN', 20000);
```

```
SQL> INSERT INTO EMPLOYEE VALUES (105, 03, 'SUMAN', 22000);
```



```
SQL> INSERT INTO EMPLOYEE VALUES (106, 02, 'SUSHMA', 18000);
SQL> INSERT INTO EMPLOYEE VALUES (109, 01, 'KUSHI', 22300);
SQL> INSERT INTO EMPLOYEE VALUES (110, 02, 'VIDHYA', 15000);
SQL> INSERT INTO EMPLOYEE VALUES (102, 02, 'KAVYA', 21300);
SQL> INSERT INTO EMPLOYEE VALUES (107, 03, 'AKASH', 18200);
SQL> INSERT INTO EMPLOYEE VALUES (108, 04, 'NAWAZ', 12000);
SQL> INSERT INTO EMPLOYEE VALUES (103, 02, 'DEEPAK', 24000);
```

To insert 4 records into the table DEPARTMENT using the INSERT INTO command.

```
SQL> INSERT INTO DEPARTMENT VALUES (01, 'PURCHASE', 'KRISHNA');
SQL> INSERT INTO DEPARTMENT VALUES (02, 'ACCOUNTS', 'TANVEER');
SQL> INSERT INTO DEPARTMENT VALUES (03, 'SALES', 'SURYA');
SQL> INSERT INTO DEPARTMENT VALUES (04, 'APPRENTICE', 'HARSHA');
```

1. Find the names of all employees who work for the Accounts department.

```
SQL> SELECT * FROM EMPLOYEE WHERE DEPT_ID=
2 (SELECT DEPT_ID FROM DEPARTMENT
3 WHERE DEPT_NAME='ACCOUNTS');
```

EMP_ID	DEPT_ID	EMP_NAME	EMP_SALARY
104	2	MOHAN	20000
106	2	SUSHMA	18000
110	2	VIDHYA	15000
102	2	KAVYA	21300
103	2	DEEPAK	24000

2. How many employees work for Accounts department?

```
SQL> SELECT COUNT(*) FROM EMPLOYEE WHERE DEPT_ID =
2 (SELECT DEPT_ID FROM DEPARTMENT
3 WHERE DEPT_NAME='ACCOUNTS');
```

COUNT(*)
5

3. What are the Minimum, Maximum and Average salary of employees working for Accounts department?

```
SQL> SELECT MIN(EMP_SALARY),
2 MAX(EMP_SALARY),
3 AVG(EMP_SALARY)
4 FROM EMPLOYEE WHERE DEPT_ID =
5 (SELECT DEPT_ID FROM DEPARTMENT
6 WHERE DEPT_NAME='ACCOUNTS');
```

MIN(EMP_SALARY)	MAX(EMP_SALARY)	AVG(EMP_SALARY)
15000	24000	19660

4. List the employees working for particular supervisor.

```
SQL> SELECT * FROM EMPLOYEE WHERE DEPT_ID =
2 (SELECT DEPT_ID FROM DEPARTMENT
3 WHERE SUPERVISOR='SURYA');
```

EMP_ID	DEPT_ID	EMP_NAME	EMP_SALARY
105	3	SUMAN	22000
107	3	AKASH	18200

5. Retrieve the department names for each department where only one employee works.

```
SQL> SELECT DEPT_NAME FROM DEPARTMENT
2 WHERE DEPT_ID IN (SELECT DEPT_ID FROM EMPLOYEE
3 GROUP BY DEPT_ID HAVING COUNT(*)=1);
```

```
DEPT_NAME
-----
APPRENTICE
```

6. Increase the salary of all employees in the sales department by 15%.

```
SQL> UPDATE EMPLOYEE
2 SET EMP_SALARY = EMP_SALARY + 15 * EMP_SALARY/100
3 WHERE DEPT_ID = (SELECT DEPT_ID FROM DEPARTMENT
4 WHERE DEPT_NAME='SALES');
```

2 rows updated.

7. Add a new Column to the table EMPLOYEE called BONUS NUMBER (5) and compute 5% of the salary to the said field.

```
SQL> ALTER TABLE EMPLOYEE ADD(BONUS NUMBER(5));
```

Table altered.

```
SQL>
SQL> UPDATE EMPLOYEE
2 SET BONUS = 5 * EMP_SALARY/100;
```

10 rows updated.

```
SQL> SELECT * FROM EMPLOYEE;
```

EMP_ID	DEPT_ID	EMP_NAME	EMP_SALARY	BONUS
101	1	ARUN	15000	750
104	2	MOHAN	20000	1000
105	3	SUMAN	25300	1265
106	2	SUSHMA	18000	900
109	1	KUSHI	22300	1115
110	2	VIDHYA	15000	750
102	2	KAUYA	21300	1065
107	3	AKASH	20930	1047
108	4	NAVAZ	12000	600
103	2	DEEPAK	24000	1200

10 rows selected.

8. Delete all the rows for the employee in the Apprentice department.

```
SQL> DELETE FROM EMPLOYEE  
2 WHERE DEPT_ID = (SELECT DEPT_ID FROM DEPARTMENT  
3 WHERE DEPT_NAME='APPRENTICE');
```

1 row deleted.



SECTION – C

ADVANCED HTML

PROGRAM 1:

Write a HTML program to create a CLASS Time Table.

```
<HTML>
<HEAD>
  <TITLE> CLASS TIME TABLE </TITLE>
</HEAD>

<BODY TEXT=DARKBLUE COLOR=WHITE>
<CENTER> <H3> M D R PU SCIENCE COLLEGE </H3>
<CENTER> <H4> TIME TABLE 2016-17 </H4>

<TABLE BORDER=10 BORDERCOLOR=RED BGCOLOR=CORNSILK CELLSPACING=2
CELLPADDING=15>
<CAPTION> <B> II PUC PCMCs </B> <CAPTION>

<TR BGCOLOR=PEACHPUFF>
  <TD ROWSPAN=2 ALIGN=CENTER> <B> DAY </B> </TD>
  <TD COLSPAN=8 ALIGN=CENTER> <B> TIMINGS </B></TD>
</TR>

<TR BGCOLOR=RED>
  <TH> 9.20 - 10.20 </TH>
  <TH> 10.20 - 11.20 </TH>
  <TH> 11.20 - 11.30 </TH>
  <TH> 11.30 - 12.30 </TH>
  <TH> 12.30 - 1.30 </TH>
  <TH> 1.30 - 2.30 </TH>
  <TH> 2.30 - 3.15 </TH>
  <TH> 3.15 - 4.00 </TH>
</TR>
<TR>
  <TD> MONDAY </TD>
  <TD> MATHS </TD>
  <TD> PHYSICS </TD>
```

	<TD ROWSPAN=6 ALIGN="CENTER">SHORT BREAK</TD>
	<TD> CHEMISTRY </TD>
	<TD> COMP SCI </TD>
	<TD ROWSPAN=6 ALIGN=CENTER>LUNCH BREAK</TD>
</TR>	<TD
<TR>	COLSPAN=2ALIGN=CENTER><.....COMP SCI LAB.....> </TD>
	<TD>TUESDAY</TD>
	<TD> PHYSICS </TD>
	<TD> MATHS </TD>
	<TD> CHEMISTRY </TD>
</TR>	<TD> ENGLISH </TD>
<TR>	<TD COLSPAN=2 ALIGN=CENTER><.....PHYSICS LAB.....></TD>
	<TD> WEDNESDAY </TD>
	<TD> COMP SCI </TD>
</TR>	<TD> PHYSICS </TD>
<TR>	<TD> MATHS </TD>
	<TD> KANNADA </TD>
	<TD COLSPAN=2 ALIGN=CENTER><.....CHEMISTRY LAB....></TD>
	<TD> THURSDAY </TD>
</TR>	<TD> MATHS </TD>
	<TD> KANNADA </TD>
	<TD> PHYSICS </TD>
	<TD> ENGLISH </TD>
	<TD COLSPAN=2 ALIGN=CENTER><.....COMP SCI LAB.....> </TD>

<TR>

<TD>FRIDAY </TD>

<TD>ENGLISH </TD>

<TD>MATHS </TD>

<TD>PHYSICS </TD>

```

<TD>COMP SCI </TD>
<TD>CHEMISTRY </TD>
<TD>KANNADA </TD>

</TR>
<TR>

<TD> SATURDAY </TD>
<TD> MATHS </TD>
<TD> ENGLISH </TD>
<TD> CHEMISTRY </TD>
<TD> COMP SCI </TD>
<TD COLSPAN=2
ALIGN=CENTER><.....SPECIAL CLASS.....>
</TD>

</TR>

</TABLE>
</CENTER>
</BODY>
</HTML>

```

OUTPUT:

M D R PU SCIENCE COLLEGE								
TIME TABLE 2016-17								
II PUC PCMCs								
DAY	TIMINGS							
	9:20 - 10:20	10:20 - 11:20	11:20 - 11:30	11:30 - 12:30	12:30 - 1:30	1:30 - 2:30	2:30 - 3:15	3:15 - 4:00
MONDAY	MATHS	PHYSICS	SHORT BREAK	CHEMISTRY	COMP SCI	LUNCH BREAK	<.....COMP SCI LAB.....>	
TUESDAY	PHYSICS	MATHS		CHEMISTRY	ENGLISH		<.....PHYSICS LAB.....>	
WEDNESDAY	COMP SCI	PHYSICS		MATHS	KANNADA		<.....CHEMISTRY LAB.....>	
THURSDAY	MATHS	KANNADA		PHYSICS	ENGLISH		<.....COMP SCI LAB.....>	
FRIDAY	ENGLISH	MATHS		PHYSICS	COMP SCI		CHEMISTRY	KANNADA
SATURDAY	MATHS	ENGLISH		CHEMISTRY	COMP SCI		<.....SPECIAL CLASS.....>	



PROGRAM 2:

Create an HTML program with Table and Form.

```
<HTML>
<HEAD>
  <TITLE> ONLINE APPLICATION </TITLE>
</HEAD>

<BODY>
<FORM NAME="APPFORMPUC" METHOD="POST" ACTION="IPUC_SEND.PHP">
<H3 ALIGN=CENTER> FIRST PUC APPLICATION FORM </H3>
<TABLE CELLSPACING=5 CELLPADDING=5% ALIGN=CENTER>
<TR>
  <TD ALIGN=LEFT>STUDENT NAME: </TD>
  <TD><INPUT TYPE="TEXT"
NAME="STUNAME"></TD>
</TR>
<TR>
  <TD ALIGN=LEFT>FATHER NAME: </TD>
  <TD><INPUT TYPE="TEXT"
NAME="FATNAME"></TD>
</TR>
<TR>
  <TD ALIGN=LEFT>FATHER OCCUPATION:
  </TD>
  <TD><INPUT TYPE="TEXT"
NAME="FATOCC"></TD>
</TR>
<TR>
  <TD ALIGN=LEFT>DATE OF BIRTH: </TD>
  <TD><INPUT TYPE="TEXT"
NAME="DOB"></TD>
</TR>
<TR>
  <TD ALIGN=LEFT>CONTACT NUMBER: </TD>
  <TD><INPUT TYPE="TEXT"
NAME="CONTACT"></TD>
</TR>
<TR>
  <TD ALIGN=LEFT> EMAIL ID: </TD>
  <TD><INPUT TYPE="TEXT"
NAME="EMAIL"></TD>
</TR>
<TR>
  <TD ALIGN=LEFT>UPLOAD PHOTO: </TD>
  <TD><INPUT TYPE=FILE
```

NAME="PHOTO"></TD>

<TD ALIGN=LEFT>GENDER: </TD>

<TD><INPUT TYPE=RADIO NAME=GEN
VALUE="M">MALE

<INPUT TYPE=RADIO NAME=GEN
VALUE="F">FEMALE</TD>

<TR>	<TD ALGIN=LEFT> CATEGORY:</TD> <TD ALGIN=LEFT><INPUT TYPE="TEXT" NAME="CATEGORY"> <SELECT NAME="DROPDOWN" > <OPTION VALUE=1>GM</OPTION> <OPTION VALUE=2>SC</OPTION> <OPTION VALUE=3>ST</OPTION> <OPTION VALUE=4>C1</OPTION> <OPTION VALUE=5>2A</OPTION> <OPTION VALUE=6>2B</OPTION> <OPTION VALUE=7>3A</OPTION> <OPTION VALUE=8>3B</OPTION> </SELECT> </TD>
</TR>	
<TR>	<TD ALIGN=LEFT>Indicate the Board PASSED: </TD> <TD ALGIN=LEFT><INPUT TYPE="TEXT" NAME="QUALIFICATION" > <SELECT NAME="DROPDOWN" SIZE=4 ID=QUALI> <OPTION VALUE=1>SSLC</OPTION> <OPTION VALUE=2>CBSE</OPTION> <OPTION VALUE=3>ICSE</OPTION> <OPTION VALUE=4>OTHER STATE</OPTION> </SELECT> </TD>
</TR>	
<TR>	<TD ALGIN=LEFT> STUDENT ADDRESS :</TD> <TD> <TEXTAREA ROWS=2 COLS=15 NAME=ADD></TEXTAREA> <P> Enter the Contact Address with Pin Code </P> </TD>
</TR>	
<TR>	<TD ALGIN=LEFT>SUBJECT CHOOSEN: </TD> <TD ALGIN=LEFT> <INPUT TYPE=CHECKBOX NAME=LANG1 >KANNADA <INPUT TYPE=CHECKBOX NAME=LANG2 >ENGLISH <INPUT TYPE=CHECKBOX NAME=SUB1 >PHYSICS

☐ SUB2 CHEMISTRY ☐ SUB3 MATHS ☐ SUB4	☐ BIOLOGY ☐ SUB5 COMP SCI ☐ ELECTRONICS
---	---

```

<TR>
    <TD><INPUT TYPE="SUBMIT"
    VALUE="SUBMIT THE FORM"> </TD>
    <TD><INPUT TYPE="RESET" VALUE="RESET
    THE FORM"></TD>
</TR>

</TABLE>
</FORM>
</BODY>
</HTML>

```

OUTPUT:

FIRST PUC APPLICATION FORM	
STUDENT NAME:	
FATHER NAME:	
FATHER OCCUPATION:	
DATE OF BIRTH:	
CONTACT NUMBER:	
EMAIL ID:	
UPLOAD PHOTO:	Choose File No file chosen
GENDER:	☒ MALE ☐ FEMALE
CATEGORY:	GM
Indicate the Board PASSED:	SSLC CBSE ICSE OTHER STATE
STUDENT ADDRESS:	
Enter the Contact Address with Pin Code	
SUBJECT CHOSEN:	☐ KANNADA ☐ ENGLISH ☐ PHYSICS ☐ CHEMISTRY ☐ MATHS ☐ BIOLOGY ☐ COMP SCI ☐ ELECTRONICS
SUBMIT THE FORM RESET THE FORM	

