II PUC COMPUTER SCIENCE LAB MANUAL

2024-2025

DEPARTMENT OF COMPUTER SCIENCE

Maharani Lakshmi Ammanni Pre-University College For Women
18th Cross Malleswaram, Bangalore-560012

Software used

HTML – notepad++
Can be installed by downloading the latest version of the file from **https://notepad-plus-plus.org/downloads/**

C++ - Turbo C++
Can be installed from **https://turbo-c.net/turbo-c-download/**

SQL – SQL Plus (command prompt version)
Can be installed from **https://www.oracle.com/database/technologies/xe-downloads.html**

PS: Please fill the index in your record book as below. Page no., Date and signature columns will be filled during lab sessions

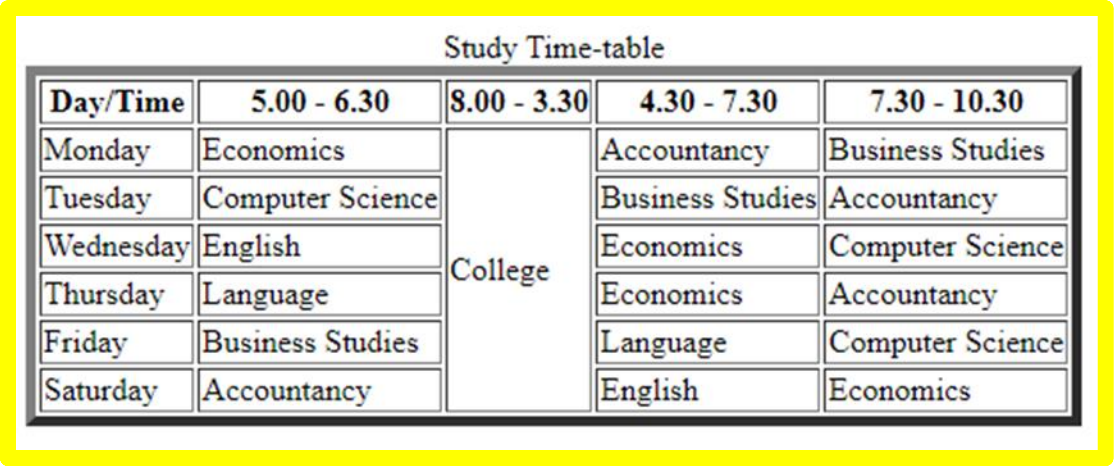
Table of Contents/Index

HTML Programs

1. Write a HTML program to create a study time-table.

```
<html>
    <head><title>Study Time-table</title></head>
     <body>
       <table border="5">
         <caption>Study Time-table</caption>
          <tr>
              <th>Day/Time
              <th>5.00 - 6.30
              <th>8.00 - 3.30
              <th>4.30 - 7.30
              <th>7.30 - 10.30
           </tr>
            <tr>
              <td>Monday
              <td>Economics
              <td rowspan="6">College
              <td>Accountancy
              <td>Business Studies
            </tr>
      <tr>
            <td>Tuesday
            <td>Computer Science
            <td>Business Studies
            <td>Accountancy
      </tr>
      <tr>
          <td>Wednesday
          <td>English
          <td>Economics
          <td>Computer Science
      </tr>
      <tr>
          <td>Thursday
          <td>Language
          <td>Economics
          <td>Accountancy
      </tr>
      <tr>
          <td>Friday
          <td>Business Studies
          <td>Language
          <td>Computer Science
      </tr>
      <tr>
          <td>Saturday
          <td>Accountancy
          <td>English
          <td>Economics
      </tr>
    </table>
  </body>
</html>
```

Output:



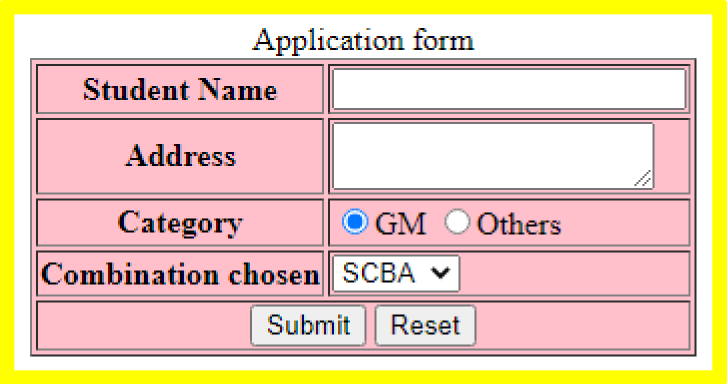
Study Time-table

Day/Time	5.00 - 6.30	8.00 - 3.30	4.30 - 7.30	7.30 - 10.30
Monday	Economics	College	Accountancy	Business Studies
Tuesday	Computer Science		Business Studies	Accountancy
Wednesday	English		Economics	Computer Science
Thursday	Language		Economics	Accountancy
Friday	Business Studies		Language	Computer Science
Saturday	Accountancy		English	Economics

2. Create an HTML program with table and Form

```
<html>
  <head><title>Application Form</title></head>
  <body>
    <form>
     <table border="1" bgcolor="pink">
       <caption>Application form</caption>

       <tr>
         <th>Student Name
          <td><input type="text">
       </tr>

       <tr>
          <th>Address
          <td><textarea></textarea>
       </tr>

       <tr>
          <th>Category
          <td>
          <input type="radio" name="cat" checked>GM
          <input type="radio" name="cat">Others
       </tr>
<tr>
   <th>Combination chosen
   <td>
   <select>
            <option>PCMC
            <option selected>SCBA
            <option>CEBA
   </select>
</tr>

<tr>
  <td colspan="2" align="center">
     <input type="submit">
     <input type="reset">
</tr>

</table>
</form>
</body>
</html>
```

Output:



Application form

Student Name	
Address	
Category	● GM ○ Others
Combination chosen	SCBA

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

```
/*To find the frequency of presence of an
element in an array*/
#include<iostream.h>
#include<conio.h>

class array
{
        private:
                int a[25],n,ele,freq;
        public:
                void getdata();
                void findfreq();
                void displaydata();
};

void array::getdata()
{
   cout<<"Enter the size of the array:";
   cin>>n;
   cout<<"Enter the elements of the array:";
   for (int i=0;i<n;i++)
        cin>>a[i];
   cout<<"Enter the element to be searched:";
   cin>>ele;
}
```

```
void array::findfreq()
{
        freq=0;
        for (int i=0;i<n;i++)
                if (a[i]==ele)
                        freq++;
}

void array::displaydata()
{
  cout<<"Element occurs  "<<freq<<"  times";
}

void main()
{
        clrscr();
        array a;
        a.getdata();
        a.findfreq();
        a.displaydata();
        getch();
}
```

Output:

```
a)  Enter the size of the array: 5
Enter the elements of the array: 2 3 5 2 4
Enter the element to be searched: 2
Element occurs 2 times

b) Enter the size of the array: 5
Enter the elements of the array: 2 3 5 2 4
Enter the element to be searched: 8
Element occurs 0 times
```

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

```
/*To insert an element into an array at a given position*/
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>

class array
{
        private:
                int a[25],n,ele,pos;
        public:
                void getdata();
                void insertdata();
                void displaydata();
};

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

```
void array::insertdata()
{
        if (pos>n)
        {
                cout<<"Invalid position ";
                getch();
                exit(0);
        }
        for (int i=n-1;i>=pos;i--)
                a[i+1]=a[i];
        a[pos]=ele;
        n=n+1;
}

void array::displaydata()
{
      cout<<"\nArray after insertion\n";
      for (int i=0;i<n;i++)
                cout<<"\t"<<a[i];
}

void main()
{
        clrscr();
        array a;
                a.getdata();
        a.insertdata();
        a.displaydata();
        getch();
}
```

Output:

```
a)  Enter the size of the array:5
Enter the elements of the array: 1 3 4 5 6
Enter the element to be inserted:2
Enter the position of insertion:1
Array after insertion
1   2   3   4   5  6

b)  Enter the size of the array:5
Enter the elements of the array: 1 3 4 5 6
Enter the element to be inserted: 2
Enter the position of insertion: 6
Invalid position
```

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

```
/*To delete an element from an array at a given
position*/
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>

class array
{
        private:
                int a[25],n,pos;
        public:
                void getdata();
                void deletedata();
                void displaydata();
};

void array::getdata()
{
    cout<<"Enter the size of the array:";
    cin>>n;
    cout<<"Enter the elements of the array:";
    for (int i=0;i<n;i++)
        cin>>a[i];
    cout<<"Enter the position of deletion:";
    cin>>pos;
}
```

```
void array::deletedata()
{
        if (pos>n-1)
        {
                cout<<"Invalid position ";
                exit(0);
        }
        for (int i=pos;i<n-1;i++)
                a[i]=a[i+1];
        n=n-1;
}

void array::displaydata()
{
      cout<<"\nArray after deletion\n";
      for (int i=0;i<n;i++)
                cout<<"\t"<<a[i];
}

void main()
{
        clrscr();
        array a;
        a.getdata();
        a.deletedata();
        a.displaydata();
        getch();
}
```

Output:

```
a)  Enter the size of the array:5
Enter the elements of the array: 1 2 3 4 5
Enter the position of deletion:3
Array after deletion
1   2   3   5

b)  Enter the size of the array:5
Enter the elements of the array:1 2 3 4 5
Enter the position of deletion:5
Invalid position
```

6. Write a program to sort the elements of an array in ascending order using Insertion Sort.

```
/*Insertion sort*/
#include<iostream.h>
#include<conio.h>

class array
{
        private:
                int a[25],n;
        public:
                void getdata();
                void sort();
                void displaydata();
};
void array::getdata()
{
    cout<<"Enter the size of the array:";
    cin>>n;
    cout<<"Enter the elements of the array:";
    for (int i=0;i<n;i++)
        cin>>a[i];
}
```

```
void array::sort()
{
    int i,j,temp;
    for (i=1;i<n;i++)
       for (j=i;j>0;j--)
            if (a[j]<a[j-1])
           temp=a[j], a[j]=a[j-1], a[j-1]=temp;
}

void array::displaydata()
{
      cout<<"\nSorted array elements\n";
      for (int i=0;i<n;i++)
          cout<<"\t"<<a[i];
}

void main()
{
        clrscr();
        array a;
        a.getdata();
        a.sort();
        a.displaydata();
        getch();
}
```

Output:

```
Enter the size of the array:5
Enter the elements of the array:5 4 2 6 1
Sorted array elements
1 2 4 5 6
```

7. Write a program to search for a given element in an array using Binary Search method.

```
/*Binary search*/
#include<iostream.h>
#include<conio.h>

class array
{
        private:
                int a[25],n,ele,loc;
        public:
                void getdata();
                void search();
                void displaydata();
};

void array::getdata()
{
        cout<<"Enter the size of the array:";
        cin>>n;
        cout<<"Enter the elements of the array:";
        for (int i=0;i<n;i++)
                cin>>a[i];
        cout<<"Enter the element to be searched:";
        cin>>ele;
}

void array::displaydata()
{
      if (loc==-1)
        cout<<"Element not found";
      else
       cout<<"Element found at location "<<loc;
}
```

```
void array::search()
{
        loc=-1;
        int b=0,e=n-1,m;
        while (b<=e)
        {
                m=(b+e)/2;
                if (ele==a[m])
                {
                        loc=m;
                        break;
                }
                if (ele<a[m])
                        e=m-1;
                else
                        b=m+1;
        }
}

void main()
{
        clrscr();
        array a;
        a.getdata();
        a.search();
        a.displaydata();
        getch();
}
```

Output:

```
a)  Enter the size of the array:5
Enter the elements of the array: 2 4 6 8 10
Enter the element to be searched:8
Element found at location 3

b)  Enter the size of the array:5
Enter the elements of the array: 2 4 6 8 10
Enter the element to be searched:1
Element not found
```

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

```
/*To calculate simple interest*/
#include<iostream.h>
#include<conio.h>

class account
{
     private:
        float p,t,r,si;
     public:
        void getdata();
        void calculatesi();
        void displaydata();
};

void account::getdata()
{
   cout<<"\nEnter principal, time and rate of interest:";
   cin>>p>>t>>r;
}

void account::calculatesi()
{
        si=p*t*r/100;
}
```

```
void account::displaydata()
{
   cout<<"\nPrincipal          :"<<p;
   cout<<"\nTime               :"<<t;
   cout<<"\nRate of interest :"<<r;
   cout<<"\nSimple interest  :"<<si;
}

void main()
{
        clrscr();
        account a;
        a.getdata();
        a.calculatesi();
        a.displaydata();
        getch();
}
```

Output:

```
Enter the principal, time and rate of interest:10000  2  5.5

Principal           :10000
Time                :2
Rate of interest    :5.5
Simple interest    :1100
```

9. **Write a 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.**
    a) **If discriminant =0, print the roots that are equal**
    b) **If the discriminant is > 0, print the real roots**
    c) **If the discriminant < 0, print that the roots are imaginary**

```
/*To find the roots of a quadratic equation */
#include<iostream.h>
#include<conio.h>
#include<math.h>
#include<stdlib.h>

class quadratic
{
        private:
                int a,b,c;
                float x1,x2;
        public:
                void getdata();
                void findroots();
                void displaydata();
};

void quadratic::getdata()
{
   cout<<"\nEnter the values of a, b, c:";
   cin>>a>>b>>c;
}

void quadratic::displaydata()
{
        cout<<"\n x1 = "<<x1;
        cout<<"\n x2 = "<<x2;
}
```

```
void quadratic::findroots()
{
    float d=b*b-4*a*c;
    if (d==0)
   {
         cout<<"\nRoots are equal";
         x1=-b/(2*a);
         x2=x1;
   }
   else if (d>0)
   {
         cout<<"\nRoots are real and distinct";
         x1=(-b+sqrt(d)) / (2*a);
         x2=(-b-sqrt(d)) / (2*a);
   }
   else
  {
         cout<<"\nRoots are imaginary";
         getch();
         exit(0);
   }
}

void main()
{
         clrscr();
         quadratic q;
         q.getdata();
         q.findroots();
         q.displaydata();
         getch();
}
```

Output:

```
a)  Enter the values of a, b, c:  1   2   1
Roots are equal
x1= -1
x2= -1

b)  Enter the values of a, b, c:  2  -3   1
Roots are real and distinct
x1=1
x2=0.5

c)  Enter the values of a, b, c: 2   -2   1
Roots are imaginary
```

10. Program to find the area of a square/rectangle/triangle using function overloading.

```
/*To find the area of
square/rectangle/triangle using
function overloading*/
#include<iostream.h>
#include<conio.h>
#include<math.h>

class shape
{
   public:
   float area(float x)
   {
      return (x*x);
   }
   float area(float l, float b)
   {
      return (l*b);
   }
   float area(float a, float b, float c)
  {
     float s=(a+b+c)/2;
     return (sqrt(s*(s-a)*(s-b)*(s-c)));
  }
};
```

```
void main()
{
  clrscr();
  shape s;
  float x, l, a, b, c;

  cout<<"\nEnter the side of the square:";
  cin>>x;
  cout<<"Area in sq. units:"<<s.area(x);

  cout<<"\nEnter length and breadth of the rectangle:";
  cin>>l>>b;
  cout<<"Area in sq.units:"<<s.area(l,b);

  cout<<"\nEnter the sides of the triangle:";
  cin>>a>>b>>c;
  cout<<"Area in sq. units:"<<s.area(a,b,c);

  getch();
}
```

Output:

```
Enter the side of the square:4
Area in sq. units:16
Enter the length and breadth of the rectangle: 4 5
Area in sq. units:20
Enter the sides of the triangle:3 4 5
Area in sq. units:6
```

11. Program to find the cube of a number using inline functions.

```
/*To find the cube of a number using inline
function*/
#include<iostream.h>
#include<conio.h>

class number
{
   public:
      inline int findcube(int a)
     {
         return (a*a*a);
     }
};
```

```
void main()
{
     clrscr();
     number n;
     int x;
     cout<<"\nEnter a number:";
     cin>>x;
     cout<<"Cube = "<<n.findcube(x);
     getch();
}
```

Output:

```
Enter a number : 6
Cube = 216
```

12. Write a program to find the sum of the series 1+ x + x^2 + … + x^n using constructors.

```
/*To find the sum of the series using constructors*/
#include<iostream.h>
#include<conio.h>
#include<math.h>

class series
{
    private:
             float x;
             int n;
    public:
           series(float b, int p)
           {
                x=b, n=p;
           }
           float sum();
};

float series::sum()
{
    float sum=1;
    for (int i=1;i<=n;i++)
            sum=sum+pow(x,i);
    return (sum);
}
```

```
void main()
{
    clrscr();
    float x;
    int n;
    cout<<"\nEnter the value of x and n:";
    cin>>x>>n;
    series s(x,n);
    cout<<"Sum of the series:"<<s.sum();
    getch();
}
```

Output:

```
Enter the value of x and n: 2   5
Sum of the series: 63
```

13. Create a base class containing the data members 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.

```
/*To show the concept of inheritance*/
#include<iostream.h>
#include<conio.h>

class student
{
    private:
      int rollno;
      char name[25];

    public:
      void getdata()
      {
        cout<<"\nEnter roll no. and name:";
        cin>>rollno>>name;
      }

      void displaydata()
     {
         cout<<"\nRoll No.:"<<rollno;
         cout<<"\nName:"<<name;
     }
};
```

```
class result : public student
{
   private:
     int m1, m2;
   public:
    void getmarks()
    {
      cout<<"\nEnter the marks in two subjects:";
      cin>>m1>>m2;
    }

    void displayresult()
    {
      cout<<"\nSub 1 marks:"<<m1;
      cout<<"\nSub 2 marks:"<<m2;
      cout<<"\nTotal :"<<m1+m2;
    }
};

void main()
{
         clrscr();
         result r;
         r.getdata();
         r.getmarks();
         r.displaydata();
         r.displayresult();
         getch();
}
```

Output:

```
Enter roll no. and name:302 Mita
Enter the marks in two subjects: 78 89
Roll No.          :302
Name              :Mita
Sub 1 Marks     :78
Sub 2 Marks     :89
Total             :167
```

14. 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.

```
/*To show concept of pointer to objects*/
#include<iostream.h>
#include<conio.h>

class student
{
        private:
                int regno;
                char name[25];
                float fees;
        public:
                void getdata();
                void displaydata();
};
void student::getdata()
{
        cout<<"Enter the register no.:";
        cin>>regno;
        cout<<"Enter the name           :";
        cin>>name;
        cout<<"Enter the fees           :";
        cin>>fees;
}
```

```
void student::displaydata()
{
        cout<<"\nStudent Details:";
        cout<<"\nRegister no.     :"<<regno;
        cout<<"\nName             :"<<name;
        cout<<"\nFees             :"<<fees;
}

void main()
{
        clrscr();
        student s, *sptr;
        sptr=&s;
        sptr->getdata();
        sptr->displaydata();
        getch();
}
```

Output:

```
Enter the register no.:245
Enter the name        :Medha
Enter the fees         :36000

Student Details:
Register no.  :245
Name          :Medha
Fees          :36000
```

15. Write a program to push items into the stack.

```
/*To show PUSH operation on stack*/
#include<iostream.h>
#include<conio.h>
#define N 4

class stack
{
        private:
                int a[N], top;
        public:
                stack();
                void push(int);
                void show();
};
stack::stack()
{
        top=-1;
}
void stack::push(int item)
{
    if (top==N-1)
        cout<<"\nStack overflow\n";
    else
    {
        top++;
        a[top]=item;
    }
}
```

```
void stack::show()
{
        if (top==-1)
                cout<<"\nStack is empty\n";
        else
        {
                cout<<"\nStack elements\n";
                for (int i=top;i>=0;i--)
                        cout<<a[i]<<endl;
        }
}
void main()
{
    clrscr();
    stack s;
    int item;

    cout<<"Enter the element to be pushed:";
    cin>>item;
    s.push(item);

    cout<<"Enter another element to be pushed:";
    cin>>item;
    s.push(item);

    s.show();
    getch();
}
```

Output:

```
Enter the element to be pushed:56
Enter another element to be pushed:28

Stack elements
28
56
```

### 16.	Write a program to pop elements from the stack.

```
/*To show POP operation on stack*/
#include<iostream.h>
#include<conio.h>
#define N 4
class stack
{
        private:
                int a[N], top;
        public:
                stack();
                void push(int);
                void pop();
                void show();
};

stack::stack()
{
        top=-1;
}
void stack::push(int item)
{
    if (top==N-1)
        cout<<"\nStack overflow\n";
   else
    {
        top++;
        a[top]=item;
     }
}

void stack::pop()
{
    if (top==-1)
        cout<<"\nStack underflow\n";
   else
   {
        int item=a[top];
        top--;
        cout<<item<<" popped from stack\n";
    }
}
```

```
void stack::show()
{
        if (top==-1)
                cout<<"\nStack is empty\n";
        else
        {
                cout<<"\nStack elements\n";
                for (int i=top;i>=0;i--)
                        cout<<a[i]<<endl;
        }
}

void main()
{
   clrscr();
   stack s;
   int item;

   cout<<"Enter the element to be pushed:";
   cin>>item;
   s.push(item);

   cout<<"Enter another element to be
pushed:";
   cin>>item;
   s.push(item);

   s.show();

   cout<<"\nAfter pop operation\n";
   s.pop();
   s.show();

   getch();
}
```

Output:

```
Enter the element to be pushed:56
Enter another element to be pushed:80
Stack elements
80
56
After pop operation
80 popped from stack

Stack elements
56
```

17. Write a program to perform enqueue and dequeue.

```
/*To perform enqueue and dequeue*/
#include<iostream.h>
#include<conio.h>
#define N 4

class queue
{
        private: int a[N], front, rear;
        public:
                queue();
                void enqueue(int);
                void dequeue();
                void show();
};

queue::queue()
{
        front = -1;
        rear = -1;
}

void queue::enqueue(int ele)
{
        if (rear==N-1)
            cout<<"\nQueue overflow\n";
        else
        {
                if (front==-1)
                        front=0,rear=0;
                else
                        rear++;
                a[rear]=ele;
        }
}
void queue::dequeue()
{
  if (front==-1)
        cout<<"\nQueue underflow\n";
  else
  {
       int ele=a[front];
        cout<<ele<<" deleted from queue\n";
        if (front==rear)
           front=-1,rear=-1;
        else
           front++;
   }
}
void queue::show()
{
        if (front==-1)
                cout<<"\nQueue is empty";
        else
        {
                cout<<"\nQueue elements\n";
                for (int i=front;i<=rear;i++)
                        cout<< "\t"<<a[i];
        }
}
void main()
{
   clrscr();
   queue q;
   int ele;
   cout<<"Enter the element to be inserted:";
   cin>>ele;
   q.enqueue(ele);
   cout<<"Enter another element to be inserted:";
   cin>>ele;
   q.enqueue (ele);
   q.show();

   cout<<"\nAfter dequeue operation\n";
   q.dequeue();
   q.show();
   getch();
}
```

Output:

```
Enter the element to be inserted:25
Enter another element to be inserted:35

Queue elements
  25  35
After dequeue operation
25 deleted from queue

Queue elements
   35
```

18. Write a program to create a linked list and appending nodes.

```
/*To create and append to a linked list*/
#include<iostream.h>
#include<conio.h>

class linkedlist
{
        private:
                struct listnode
                {
                        int data;
                        listnode *link;
                };
                listnode *head;
        public:
                linkedlist();
                void append(int);
                void show();
};

linkedlist::linkedlist()
{
        head=NULL;
}

void linkedlist::append(int info)
{
        listnode *newnode, *currptr;
        newnode=new listnode;
        newnode->data=info;
        newnode->link=NULL;
        if (head==NULL)
                head=newnode;
        else
        {
                currptr=head;
                while (currptr->link!=NULL)
                        currptr=currptr->link;
                currptr->link=newnode;
        }
}
```

```
void linkedlist::show()
{
   if (head==NULL)
      cout<<"Linked list is empty";
   else
  {
       cout<<"\nLinked list contents\n";
       listnode *currptr=head;
      while (currptr!=NULL)
     {
         cout<<"  "<<currptr->data;
         currptr=currptr->link;
      }
   }
}

void main()
{
     clrscr();
     linkedlist list;
     int ele;
     cout<<"Enter the element to be appended:";
     cin>>ele;
     list.append(ele);

     cout<<"Enter another element to be appended:";
     cin>>ele;
     list.append(ele);

     list.show();
     getch();
}
```

Output:

```
Enter the element to be appended:22
Enter another element to be appended:33
Linked list contents
 22  33
```

SQL Programs

19. Generate the Electricity Bill for one consumer.

a) Create a table for house hold electricity bill with the following fields:

Field Name	Type	Description
RR_Number	varchar2(10)	Revenue Register No.
Consumer_name	varchar2(25)	Consumer Name
date_billing	date	Billing Date
units	number(4)	Units Consumed

b) Insert 10 records into the table
c) Check the structure of the table
d) Add two new fields in the table, bill_amt number(6,2) and due_date date
e) Compute the bill amount for each customer as per the following rules

i.	min_amt	Rs.50/-
ii.	First 100 units	Rs. 4.50 / unit
iii.	> 100 units	Rs. 5.50 /unit

f) Compute due date as billing date + 15 days
g) List all the bills generated

Solution:

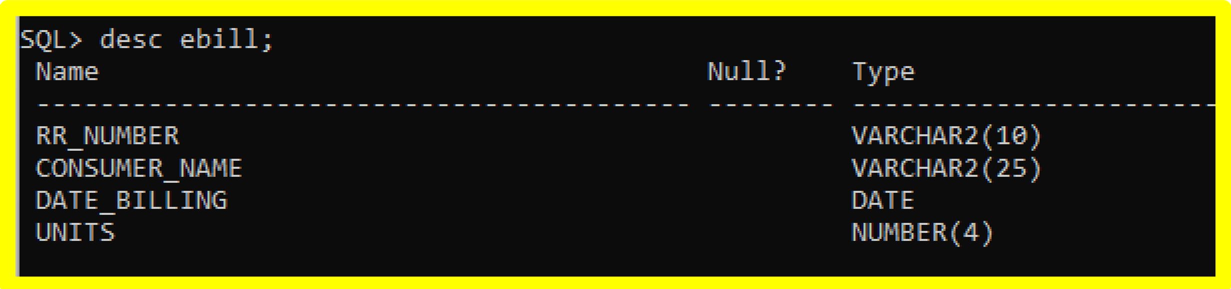
a) To create the required table,

create table ebill (RR_Number varchar2(10), Consumer_name varchar2(25), date_billing date, units number(4));

b) To insert 10 records into the table,

**insert into ebill values('eh1001','Aditi','01-apr-2024',98);
insert into ebill values('eh1003','Arun','01-apr-2024',108);
insert into ebill values('eh1010','Sangeetha','05-apr-2024',128);
insert into ebill values('eh1210','Arun','05-apr-2024',88);
insert into ebill values('eh1018','Babitha','02-apr-2024',78);
insert into ebill values('eh2010','Ram','02-apr-2024',120);
insert into ebill values('eh2020','Sita','03-apr-2024',200);
insert into ebill values('eh3100','Gita','04-apr-2024',60);
insert into ebill values('eh3010','Gopal','06-apr-2024',95);
insert into ebill values('eh3210','Raj','07-apr-2024',150);**

c) To check the structure of the table,
desc ebill;



```
SQL> desc ebill;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 RR_NUMBER                                          VARCHAR2(10)
 CONSUMER_NAME                                      VARCHAR2(25)
 DATE_BILLING                                       DATE
 UNITS                                              NUMBER(4)
```

d) To add two new fields, namely bill_amt of type number(6,2) and due_date of type date,

alter table ebill add (bill_amt number(6,2), due_date date);

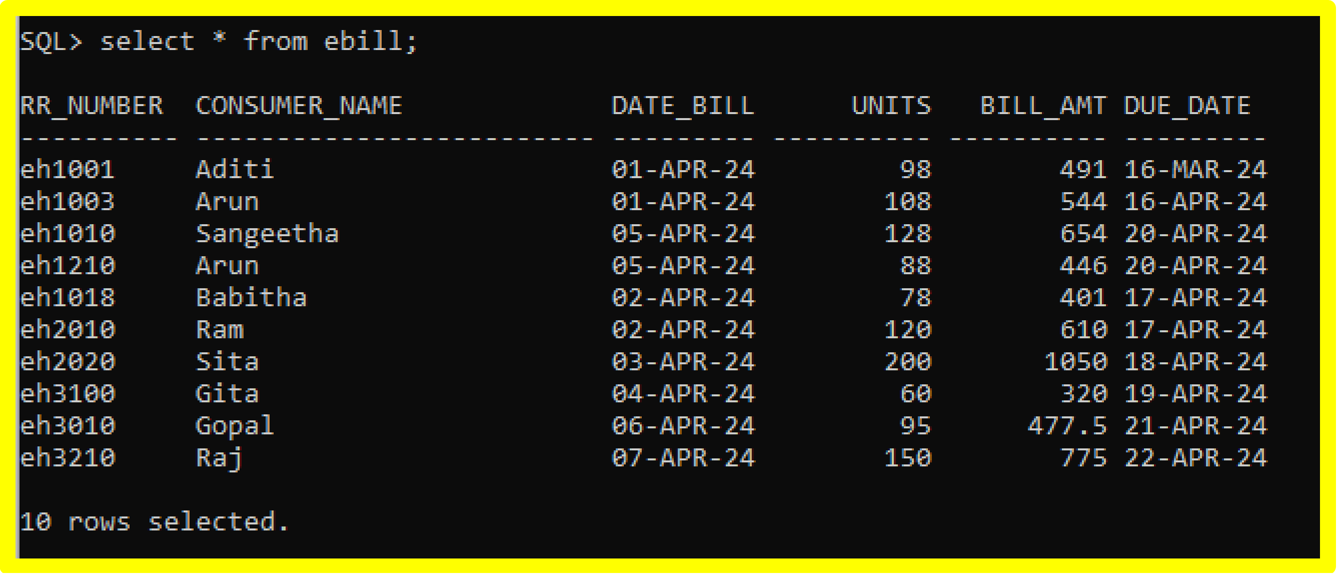
e) To compute the bill amount for each customer as per the rules,
update ebill set bill_amt=50;
update ebill set bill_amt=bill_amt + units * 4.50 where units <= 100;
update ebill set bill_amt=bill_amt+ 100*4.50 + (units-100)* 5.50 where units>100;

f) To update due date to billing date+15 days,

update ebill set due_date =date_billing+15;

g) To list all the bills generated,

select * from ebill;



```
SQL> select * from ebill;

RR_NUMBER  CONSUMER_NAME             DATE_BILL      UNITS   BILL_AMT DUE_DATE
---------- ------------------------- --------- ---------- ---------- ---------
eh1001     Aditi                     01-APR-24         98        491 16-MAR-24
eh1003     Arun                      01-APR-24        108        544 16-APR-24
eh1010     Sangeetha                 05-APR-24        128        654 20-APR-24
eh1210     Arun                      05-APR-24         88        446 20-APR-24
eh1018     Babitha                   02-APR-24         78        401 17-APR-24
eh2010     Ram                       02-APR-24        120        610 17-APR-24
eh2020     Sita                      03-APR-24        200       1050 18-APR-24
eh3100     Gita                      04-APR-24         60        320 19-APR-24
eh3010     Gopal                     06-APR-24         95      477.5 21-APR-24
eh3210     Raj                       07-APR-24        150        775 22-APR-24

10 rows selected.
```

20. Create a student database and compute the result.

a) Create a table for a class of students with the following details.

Field Name	Type	Description
Student_id	number(4)	Student id
student_name	varchar2(25)	Student name
sub1_marks	number(2)	Subject 1 marks
sub2_marks	number(2)	Subject 2 marks
sub3_marks	number(2)	Subject 3 marks
sub4_marks	number(2)	Subject 4 marks
sub5_marks	number(2)	Subject 5 marks
sub6_marks	number(2)	Subject 6 marks

b) Add records into the table for 10 students for the above-mentioned fields using INSERT command
c) Display the description of the fields in the table

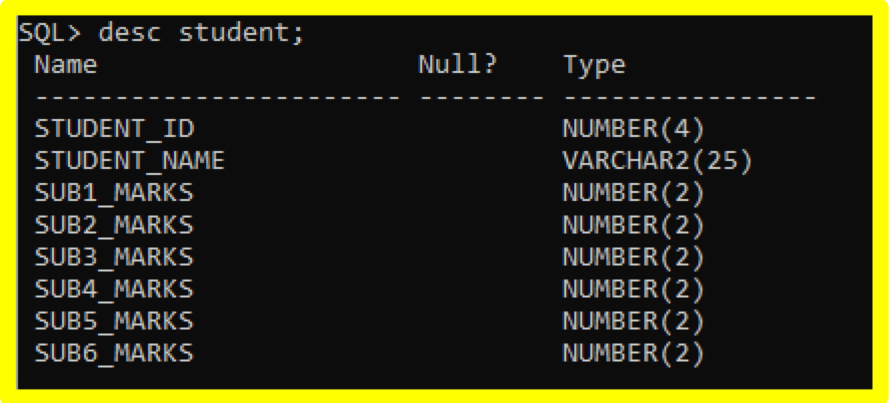
d) Alter the table and calculate total and perc_marks
e) Compute the result as “PASS” or “FAIL” by checking if the student has scored greater than or equal to 35 marks in each subject
f) Retrieve only student_id and student_name of all the students
g) List the students who have result as “PASS”
h) Count the number of students who have failed
i) List the students who have percentage greater than 60
j) Sort the table according to the order of student id

Solution:

a) To create the table STUDENT,
create table student (Student_id number(4), student_name varchar2(25), sub1_marks number(2), sub2_marks number(2), sub3_marks number(2), sub4_marks number(2), sub5_marks number(2), sub6_marks number(2));

b) To insert 10 records into the table,
insert into student values (1003, 'Arya', 56,67,77,86,98,88);
insert into student values (1002, ‘Jay’, 28,30,45,40,44,35);
insert into student values (1004, ‘Vismay’, 30,30,28,15,32,45);
insert into student values (1001, ‘Zeba’, 66,75,67,89,90,77);
insert into student values (1005, ‘Tom’, 70,80,87,89,90,97);
insert into student values (1007, ‘Jeevika’, 60,50,75,76,55,81);
insert into student values (1009, ‘Deeksha’, 45,56,35,57,70,39);
insert into student values (1006, ‘Babitha’, 27,30,35,46,60,56);
insert into student values (1008, ‘Chinmay’, 66,40,35,45,66,61);
insert into student values (1011, ‘Ira’, 50,51,63,72,67,54);

c) To display the description of the fields in the table,
desc student;



```
SQL> desc student;
 Name                       Null?    Type
 -------------------------- -------- ----------------
 STUDENT_ID                          NUMBER(4)
 STUDENT_NAME                        VARCHAR2(25)
 SUB1_MARKS                          NUMBER(2)
 SUB2_MARKS                          NUMBER(2)
 SUB3_MARKS                          NUMBER(2)
 SUB4_MARKS                          NUMBER(2)
 SUB5_MARKS                          NUMBER(2)
 SUB6_MARKS                          NUMBER(2)
```

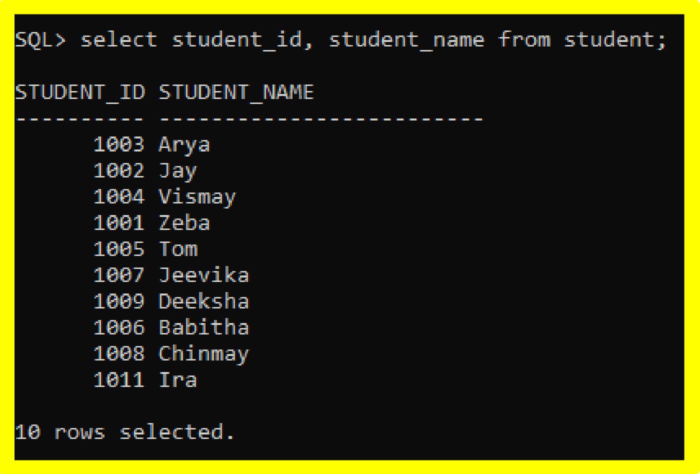
d) To add 3 columns, total as number(3), perc_marks as number(6,2) and result as varchar2(10),
alter table student add (total number(3), perc_marks number(6,2), result varchar2(10));

To compute total mark,
update student set total =sub1_marks+sub2_marks+sub3_marks+sub4_marks+sub5_marks+sub6_marks;
To compute the percentage,
update student set perc_marks=total/6.0;
e) To compute the result as “PASS” or “FAIL”,

update student set result='pass' where sub1_marks>=35 and sub2_marks>=35 and sub3_marks>=35 and sub4_marks>=35 and sub5_marks>=35 and sub6_marks>=35;

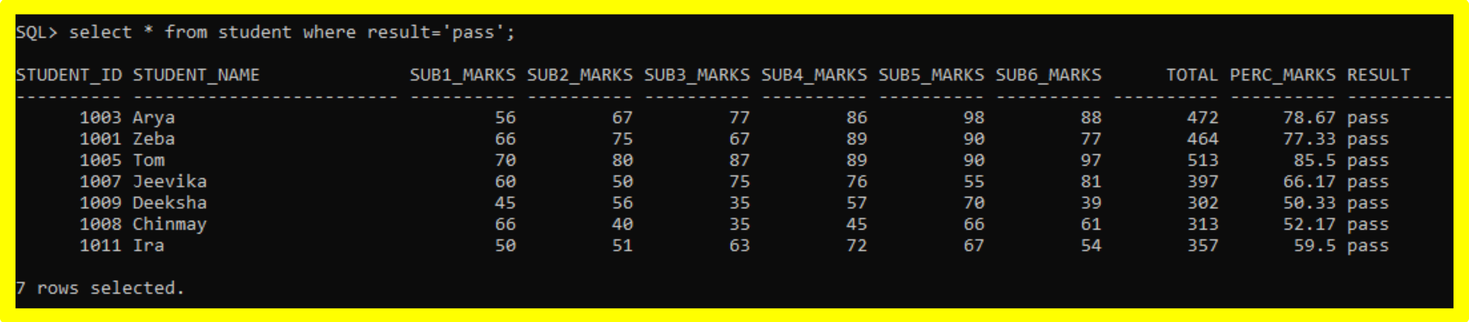
update student set result='fail' where sub1_marks<35 or sub2_marks<35 or sub3_marks<35 or sub4_marks<35 or sub5_marks<35 or sub6_marks<35;

f) To retrieve only student_id and student_name of all students,
select student_id, student_name from student;



```
SQL> select student_id, student_name from student;

STUDENT_ID STUDENT_NAME
---------- -------------------------
      1003 Arya
      1002 Jay
      1004 Vismay
      1001 Zeba
      1005 Tom
      1007 Jeevika
      1009 Deeksha
      1006 Babitha
      1008 Chinmay
      1011 Ira

10 rows selected.
```

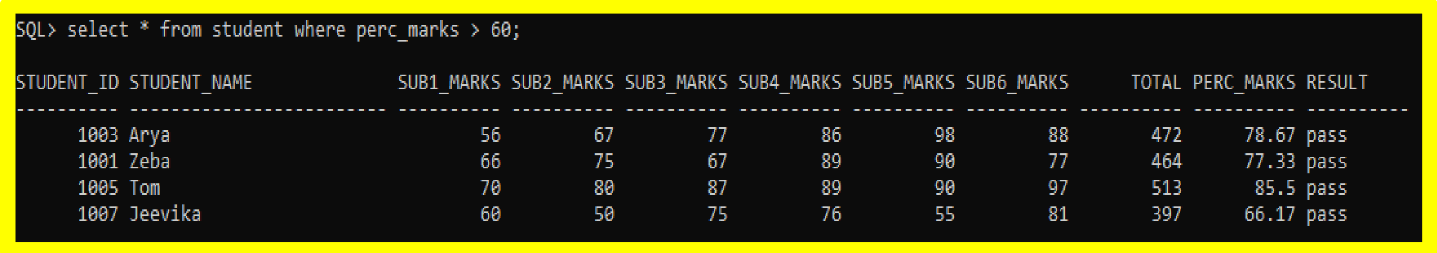
g) To list the students who have result as "PASS"
select * from student where result='pass';



```
SQL> select * from student where result='pass';

STUDENT_ID STUDENT_NAME              SUB1_MARKS SUB2_MARKS SUB3_MARKS SUB4_MARKS SUB5_MARKS SUB6_MARKS      TOTAL PERC_MARKS RESULT
---------- ------------------------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
      1003 Arya                              56         67         77         86         98         88        472      78.67 pass
      1001 Zeba                              66         75         67         89         90         77        464      77.33 pass
      1005 Tom                               70         80         87         89         90         97        513       85.5 pass
      1007 Jeevika                           60         50         75         76         55         81        397      66.17 pass
      1009 Deeksha                           45         56         35         57         70         39        302      50.33 pass
      1008 Chinmay                           66         40         35         45         66         61        313      52.17 pass
      1011 Ira                               50         51         63         72         67         54        357       59.5 pass

7 rows selected.
```

h) To count the number of students who have failed
select count(*) from student where result='fail';

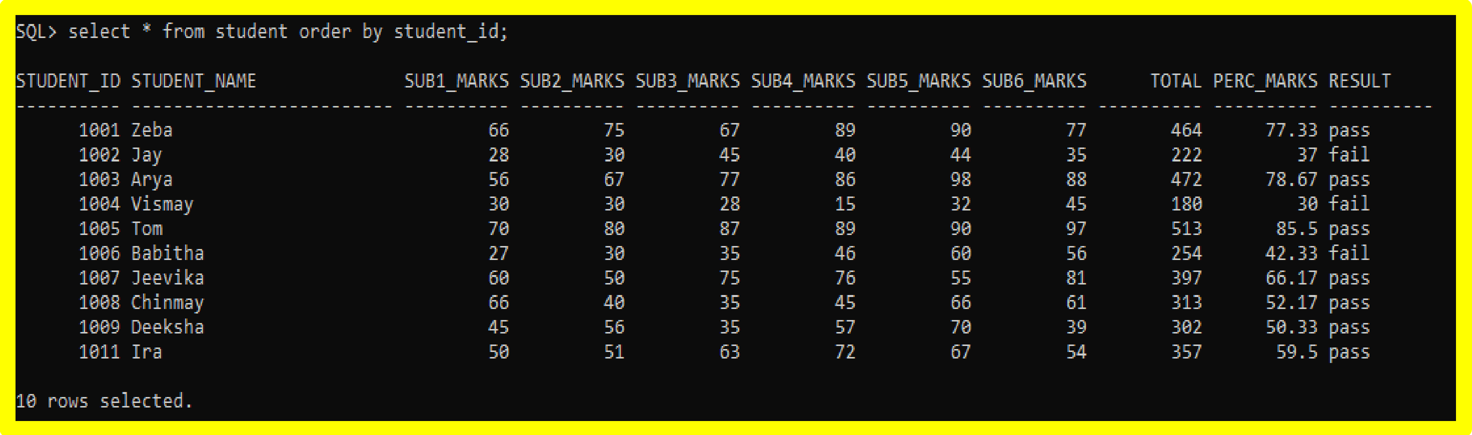
i) To list the students who have percentage greater than 60
select * from student where perc_marks > 60;



```
SQL> select * from student where perc_marks > 60;

STUDENT_ID STUDENT_NAME              SUB1_MARKS SUB2_MARKS SUB3_MARKS SUB4_MARKS SUB5_MARKS SUB6_MARKS      TOTAL PERC_MARKS RESULT
---------- ------------------------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
      1003 Arya                              56         67         77         86         98         88        472      78.67 pass
      1001 Zeba                              66         75         67         89         90         77        464      77.33 pass
      1005 Tom                               70         80         87         89         90         97        513       85.5 pass
      1007 Jeevika                           60         50         75         76         55         81        397      66.17 pass
```

j) To sort the table according to the order of STDID,

select * from student order by student_id;



```
SQL> select * from student order by student_id;

STUDENT_ID STUDENT_NAME              SUB1_MARKS SUB2_MARKS SUB3_MARKS SUB4_MARKS SUB5_MARKS SUB6_MARKS      TOTAL PERC_MARKS RESULT
---------- ------------------------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
      1001 Zeba                              66         75         67         89         90         77        464      77.33 pass
      1002 Jay                               28         30         45         40         44         35        222         37 fail
      1003 Arya                              56         67         77         86         98         88        472      78.67 pass
      1004 Vismay                            30         30         28         15         32         45        180         30 fail
      1005 Tom                               70         80         87         89         90         97        513       85.5 pass
      1006 Babitha                           27         30         35         46         60         56        254      42.33 fail
      1007 Jeevika                           60         50         75         76         55         81        397      66.17 pass
      1008 Chinmay                           66         40         35         45         66         61        313      52.17 pass
      1009 Deeksha                           45         56         35         57         70         39        302      50.33 pass
      1011 Ira                               50         51         63         72         67         54        357       59.5 pass

10 rows selected.
```

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

a) Create the following table **employee**

Field name	Data type	Description
emp_id	number(4)	employee's id number
dept_id	number(2)	department's id number
emp_name	varchar2(25)	employee name
emp_salary	number(5)	salary of the employee

b) Create another table **department** with the following structure

Field name	Data type	Description
dept_id	number(2)	department's id number
dept_name	varchar2(20)	Department name
supervisor	varchar2(20)	Department head's name

Assume the department names as purchase (id-01), accounts (id-02), sales (id-03) and apprentice (id-04).

c) Enter 10 rows of data for table **EMPLOYEE** and 4 rows of data for **DEPARTMENT** table

Write SQL statements for the following:

d) Find the names of all employees who work for the accounts department

e) What is the minimum, maximum and average salary of employees working for accounts department?

f) Increase the salary of all employees in the sales department by 15%

g) Add a new column to the table employee, called Bonus number(5) and compute 5% of the salary to the said field

h) Delete all the rows for employees in the apprentice department

Solution:

a) To create the table **employee**,
create table employee (emp_id number(4), emp_name varchar2(25), dept_id number(2), emp_salary number(5));

b) To create the table **department**,
create table department (dept_id number(2), dept_name varchar2(20), supervisor varchar2(20));

c) To insert the department details for 4 departments,

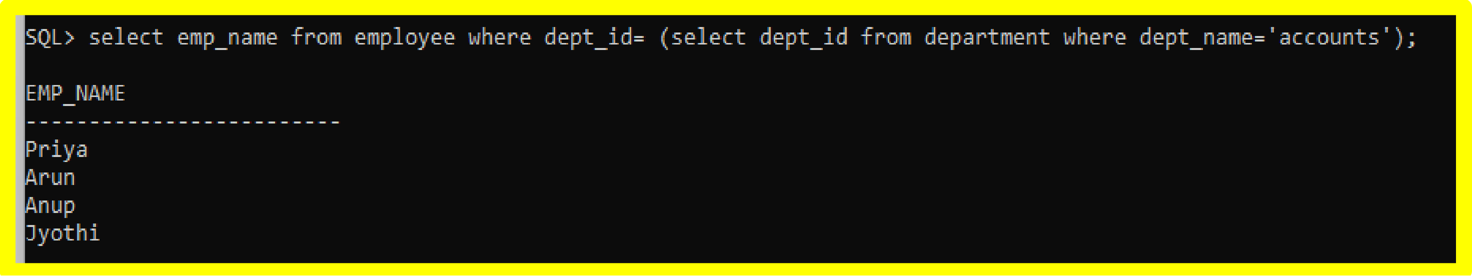
insert into department values (01,'purchase','Harsha');
insert into department values (02,'accounts','Priya');
insert into department values (03,'sales','Ashwin');
insert into department values (04,'apprentice','Hari');

To insert 10 records into **employee** table,
The first 4 records are the records of supervisors as they too are employees.

insert into employee values (1029,'Harsha',01,28900);
insert into employee values (1012,'Priya',02,28000);
insert into employee values (1001,'Ashwin',03,36500);
insert into employee values (1034,'Hari',04,37000);

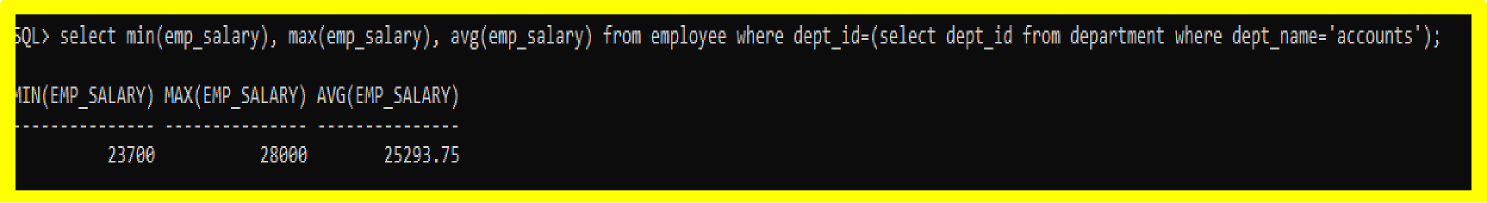
insert into employee values (1001,’Arif’,03,24150);
insert into employee values (1015,’Arun’,02,25000);
insert into employee values (1028,’Anup’,02,23700);
insert into employee values (1044,’Anil’,04,19350);
insert into employee values (1039,’Keerthi’,03,23575);
insert into employee values (1051,’Jyothi’,02,24475);

d) Find the names of all employees who work for the accounts department
select emp_name from employee where dept_id= (select dept_id from department where dept_name='accounts');



```
SQL> select emp_name from employee where dept_id= (select dept_id from department where dept_name='accounts');

EMP_NAME
-------------------------
Priya
Arun
Anup
Jyothi
```

e) To find the minimum, maximum and average salary of employees who work for the accounts department,
select min(emp_salary), max(emp_salary), avg(emp_salary) from employee where dept_id=(select dept_id from department where dept_name='accounts');

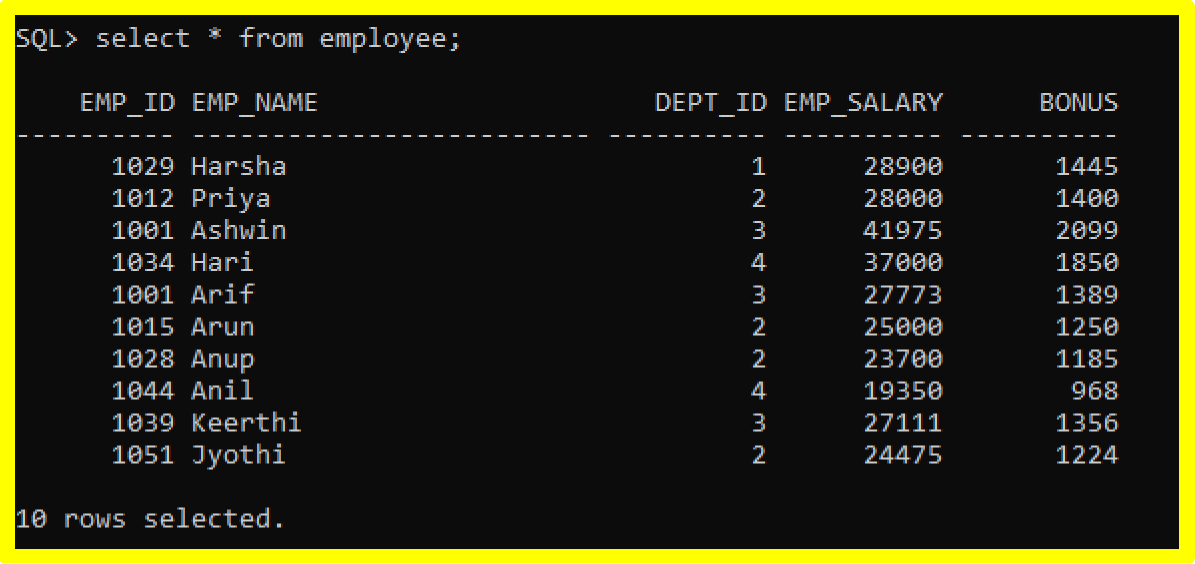


```
SQL> select min(emp_salary), max(emp_salary), avg(emp_salary) from employee where dept_id=(select dept_id from department where dept_name='accounts');

MIN(EMP_SALARY) MAX(EMP_SALARY) AVG(EMP_SALARY)
--------------- --------------- ---------------
          23700           28000        25293.75
```

f) To increase the salary of all employees in the sales department by 15%,
update employee set emp_salary=emp_salary+0.15*emp_salary where dept_id=(select dept_id from department where dept_name='sales');

g) To add a new column BONUS of type int in the table EMPLOYEE,
alter table employee add (bonus number(5));
To compute 5% of EMPSAL as BONUS for each employee,
update employee set bonus=0.05*emp_salary;
To check the changes to the EMPLOYEE table,

select * from employee;

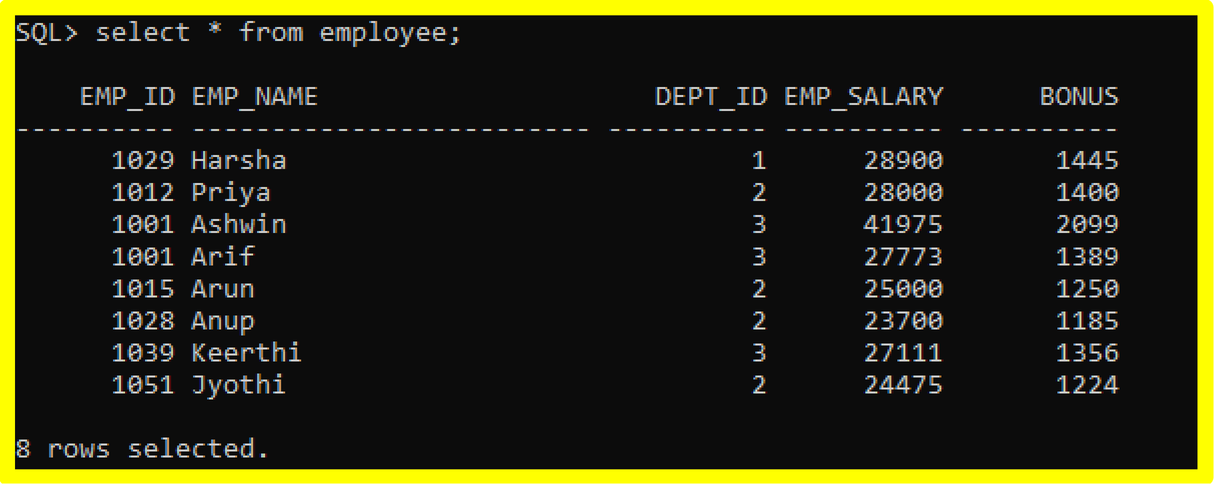


```
SQL> select * from employee;

    EMP_ID EMP_NAME                                DEPT_ID EMP_SALARY      BONUS
---------- ------------------------- ---------- ---------- ----------
      1029 Harsha                                        1      28900       1445
      1012 Priya                                         2      28000       1400
      1001 Ashwin                                        3      41975       2099
      1034 Hari                                          4      37000       1850
      1001 Arif                                          3      27773       1389
      1015 Arun                                          2      25000       1250
      1028 Anup                                          2      23700       1185
      1044 Anil                                          4      19350        968
      1039 Keerthi                                       3      27111       1356
      1051 Jyothi                                        2      24475       1224

10 rows selected.
```

h) To delete all rows for employees in the APPRENTICE department,
delete from employee where dept_id =
(select dept_id from department where dept_name='apprentice');

The result can be viewed by using the command,
select * from employee;

This returns a table where no rows have dept_id=04



```
SQL> select * from employee;

    EMP_ID EMP_NAME                                DEPT_ID EMP_SALARY      BONUS
---------- ------------------------- ---------- ---------- ----------
      1029 Harsha                                        1      28900       1445
      1012 Priya                                         2      28000       1400
      1001 Ashwin                                        3      41975       2099
      1001 Arif                                          3      27773       1389
      1015 Arun                                          2      25000       1250
      1028 Anup                                          2      23700       1185
      1039 Keerthi                                       3      27111       1356
      1051 Jyothi                                        2      24475       1224

8 rows selected.
```

22. Create database for the bank transaction.

a) Create table **customer** with the following fields.

Field Name	Data type	Description
cno (Primary key)	number(4)	Customer number
cname	char(25)	Customer name
caddress	varchar2(50)	Customer address
cphone	char(12)	Customer phone

b) Create table **bank** with the following fields.

Field Name	Data type	Description
accno (not null)	number(8)	Account number
tramount (> 0)	number(8,2)	Transaction Amount
trdate	date	Transaction Date
trtype(d-debit,c-credit)	char	Transaction type
cno	number(4)	Customer number

c) List all transactions that have been carried out between specific dates
d) To get the details of customers who have done transactions by joining the two tables using a common field
e) Get the no. of transactions of each type
f) Get today’s date using dual table
g) Get the records of customers whose address has a specific pattern.
h) Get distinct customer numbers of customers who have performed some transaction from BANK table
i) Display all records from customer table whose phone number is NULL
j) Delete all records from **customer** table
k) Delete **bank** table

Solution:

a) To create the **customer** table,

```
create table customer (cno number(4) primary key, cname char(25), caddress
varchar(50), cphone char(12));
```

To insert records into CUSTOMER table,

```
insert into customer values (101, 'Reema', '18th cross, malleswaram,
bangalore',’9341232145’);
insert into customer values (102, 'Samarth', 'mathikere, bangalore',’7645798123’);
insert into customer values (103, 'Jayanth', '4th cross yeshwanthpur,
bangalore',’8812343125’);
insert into customer values (104, 'Akash', '4th cross mathikere, bangalore',NULL);
```

To check PRIMARY KEY constraint, we use insert command,

```
insert into customer values (103, 'Aditi', '4th cross r t nagar, bangalore',’9981243135’);
```

This gives an error stating unique constraint violated.

b) To create the BANK table,

```
create table bank (accno number(8) not null, tramount number(8,2)
check(tramount>0), trdate date, trtype char, cno number(4));
```

To insert records into BANK table,

```
insert into bank values (12601,25000,'01-apr-2024','d',101);
insert into bank values (12610,5000,'25-apr-2024','c',101);
insert into bank values (12501,2400,'15-may-2024','c',102);
insert into bank values (12450,59085.50,'23-may-2024','c',103);
```

To check NOT NULL constraint, we use the command,

```
insert into bank values (NULL,25000,'01-aug-2024','d',101);
```

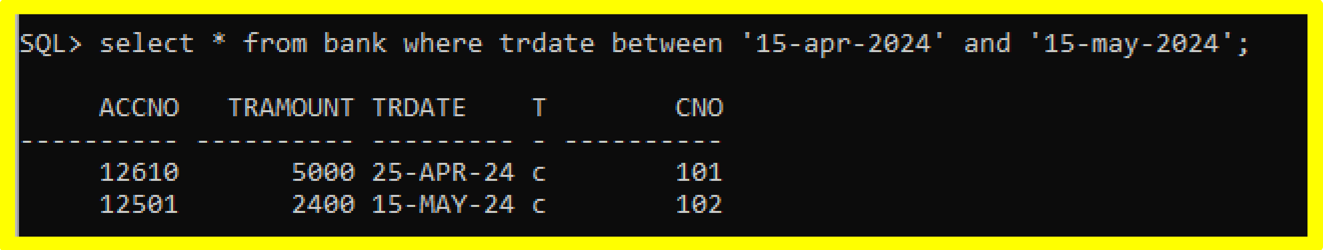
This gives an error stating the “cannot insert NULL into ("SYSTEM"."BANK"."ACCNO")”

To check CHECK constraint, we use the command,

insert into bank values (12620,0,'01-sep-2024','d',103);

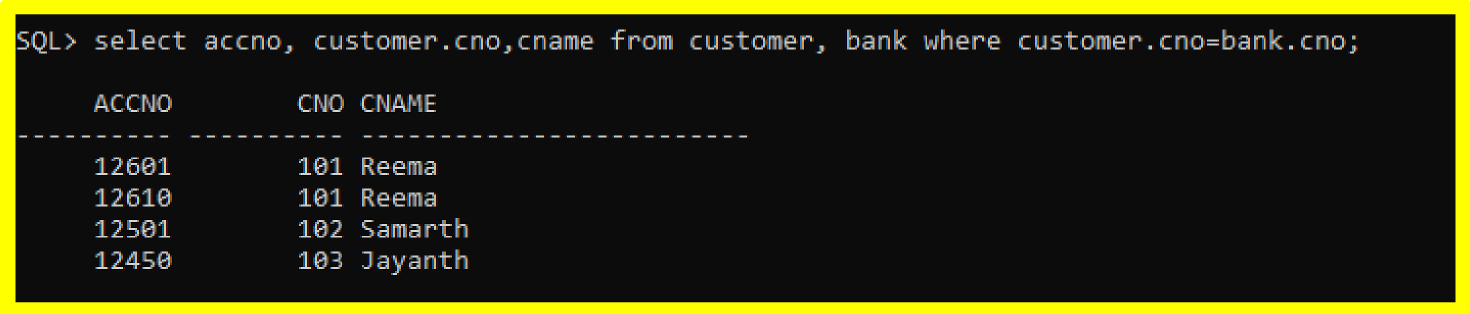
This gives an error stating the CHECK constraint violated.

c) To list all transactions with transaction date between two dates,
select * from bank where trdate between '15-apr-2024' and '15-may-2024';



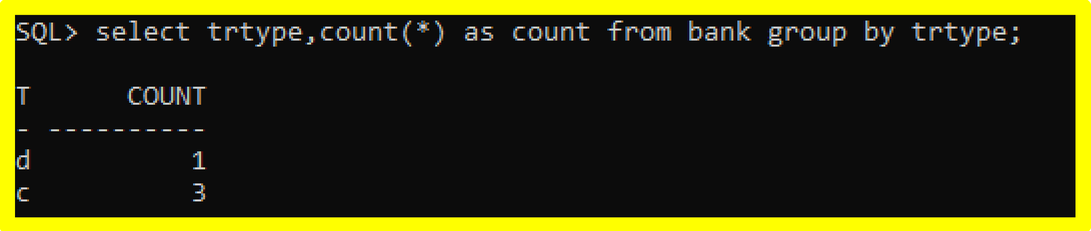
```
SQL> select * from bank where trdate between '15-apr-2024' and '15-may-2024';

     ACCNO   TRAMOUNT TRDATE    T        CNO
---------- ---------- --------- - ----------
     12610       5000 25-APR-24 c        101
     12501       2400 15-MAY-24 c        102
```

d) To get the details of customers who have done transactions by joining the two tables using a common field
select accno, customer.cno,cname from customer, bank where customer.cno=bank.cno;



```
SQL> select accno, customer.cno,cname from customer, bank where customer.cno=bank.cno;

     ACCNO        CNO CNAME
---------- ---------- -------------------------
     12601        101 Reema
     12610        101 Reema
     12501        102 Samarth
     12450        103 Jayanth
```

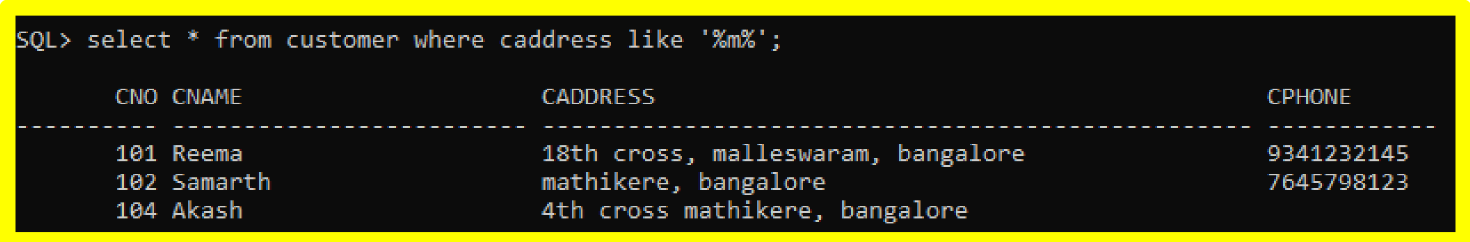
e) To get the no. of transactions of each type,
select trtype,count(*) as count from bank group by trtype;



```
SQL> select trtype,count(*) as count from bank group by trtype;

T      COUNT
- ----------
d          1
c          3
```

f) To get today's date using dual table
select sysdate from dual;

g) To list all details of customers whose address has 'M',
select * from customer where caddress like '%m%';

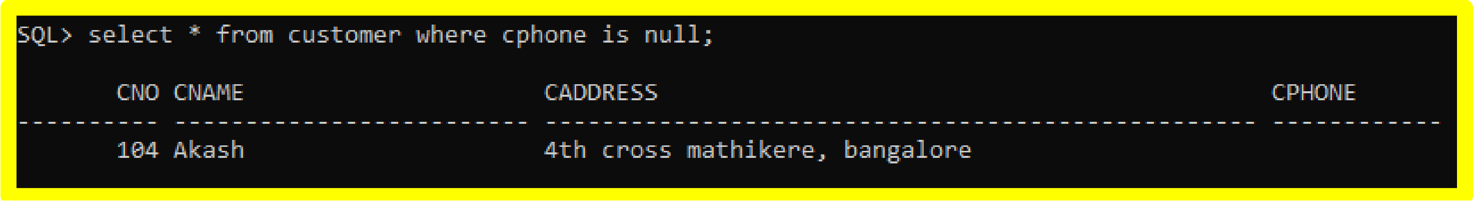


```
SQL> select * from customer where caddress like '%m%';

       CNO CNAME                     CADDRESS                                           CPHONE
---------- ------------------------- -------------------------------------------------- ------------
       101 Reema                     18th cross, malleswaram, bangalore                 9341232145
       102 Samarth                   mathikere, bangalore                               7645798123
       104 Akash                     4th cross mathikere, bangalore
```

h) In order to get the list of distinct customer numbers,

select distinct(cno) from bank;

i) Display all records from **customer** table whose phone number is NULL
select * from customer where cphone is null;



```
SQL> select * from customer where cphone is null;

       CNO CNAME                     CADDRESS                                           CPHONE
---------- ------------------------- -------------------------------------------------- ------------
       104 Akash                     4th cross mathikere, bangalore
```

j) To delete all rows in **customer** table,
delete from customer;

k) To delete the table **bank**,
drop table bank;

Viva questions

Review of C++

1. What is the file extension of a C++ program?
   .cpp

2. Why do we need
   #include<iostream.h>
   The definition of the basic input and output functionalities are defined in the iostream header file. So header file needs to be included.
   #include<conio.h>
   Required for console input/output operations, specially for clrscr() function
   #include<stdlib.h>
   For using exit() function
   #include<math.h>
   For using mathematical functions like sqrt(), pow()

3. Define keyword.
   Predefined words with some meaning and must be used only for the purpose for which they have been defined.

4. Mention the built-in data types.
   Integers (int), Characters (char), Floating point (float) and double.

5. What is the purpose of cin/cout object?
   *cin (console input) object is used with **extraction operator >>**. During the execution of the program the extraction operator extracts value to the variable in the input stream from the user.*
   *cout (console output) object is used with **insertion operator <<**. During the execution of the program the insertion operator inserts a value onto the output stream.*

6. What does endl do?
   It directs the cursor to the new line. It is a manipulator (helper function that helps to control output stream)

7. What is the function of exit ()?
   Terminate the execution of a program.

8. Which header file contains exit () function declaration?
   stdlib.h

9. Which is the header file used for mathematical functions or Name the header file used for sqrt() function?
   math.h

10. Define an array.
    Ordered collection of elements of the same type and same name.

11. What is data type of the array subscript?
    Integer.

12. What is the subscript of the first element of the array?
    Zero

13. Name the different types of arrays.
One-dimensional, Two-dimensional, Multi-dimensional.

14. What is a user-defined function?
User-defined function is a function defined by the user to perform a specific task.

15. What does the keyword void represent?
*Functions declared as **void** do not return a value.*

16. What type of value the main () function returns, by default?
Integer.

17. What is a function prototype?
Declaration of functions that tells the type of the value returned by the function and the number and type of arguments.

<u>Data Structures</u>

1. What are data structures?
A data structure is a specialized format for organizing and storing data.

2. How are data structures classified?
Data structures are classified as Primitive and Non-primitive data structures.

3. What are primitive data structures?
Data structures that are directly operated upon by machine –level instructions.

4. Give any two examples for primitive data structures.
Integer, real (float), logical data, character data, pointer and reference.

5. Mention any two operations that can be performed on primitive data structures.
Create, Destroy, Select, Update

6. What are non-primitive data structures?
Data structures that are derived from the primitive data structures and that cannot be manipulated directly by machine instructions

7. Give any two examples for non-primitive data structures.
Arrays, Stacks, Queues, Linked Lists, Trees, Graphs

8. How are non-primitive data structures classified?
They are classified as Linear and Non-linear data structures.

9. What are linear data structures? Give example.
Linear data structures are a kind of data structure that has homogeneous elements with a linear relationship between them (relationship of adjacency).
Arrays, Stacks, Queues and Linked lists are examples.

10. What are non-linear data structures? Give example.

A non-linear data structure is a data structure in which a data item is connected to several other data items but the data items are not arranged in a sequential structure.
Trees and Graphs are examples.

11. Basic operations on linear data structure.
Traversal: Process of accessing each data item exactly once to perform some operation
Insertion: Process of adding a new data item into the given collection of data items
Deletion: Process of removing an existing data item from the given collection of data items
Searching: Process of finding the location of a data item in the given collection of data items
Sorting: Process of arranging the data items in ascending or descending order
Merging: Process of combining the data items of two similar data structures to form a new data structure of same type

12. What is base address in an array?
Base address is the address of the first element of a linear array

13. What is meant by frequency of presence of an element in an array? (Prg 1)
Frequency refers to the number of times a given element occurs in an array

14. Which operation of an array does the frequency program represent? (Prg 1)
Searching / Linear searching operation

15. When is it that we cannot insert a new element into an array of n elements? (Prg 2)
When the inserting position is greater than n (OR) pos>n

16. Will the size of the array increases or decreases in insertion program? (Prg 2)
Increases

17. What is the size of an array after inserting an element? (Prg 2)
If the original size is N, after inserting an element it will be N+1

18. What is the size of an array after deleting an element? (Prg 3)
If the original size is N, after deleting an element it will be N-1

19. What is insertion sort? (OR) How insertion sort works? (Prg 4)
Every time, one element is taken from the unsorted part and is inserted in the sorted list. Hence the name, insertion sort.

20. Mention any two methods that can be used to sort the elements in an array. (Prg 4)
Bubble sort, Insertion sort, Merge sort, Quick sort, Heap sort

21. Mention the two most common methods of searching in an array. (Prg 5)
Linear search and Binary search

22. How should be the elements of the array arranged to perform the operation of binary search? (Prg 5)
The elements must be in sorted order

23. When is it not possible to perform the operation of binary search? (Prg 5)
When array is not sorted, it is not possible to perform binary search

24. What is the formula used to find the middle element of an array in binary search? (Prg 5)
m = (LB + UB) / 2 where LB – lower bound and UB – upper bound

25. If the search element is greater than the middle element, then which part of the array is searched? (Prg 5)
The second (right) part of the array (i.e., elements after middle element).

26. What is a stack?
An ordered collection of items where the addition of new items and deletion of existing items always takes place at the same end called the top of the stack

27. What is the ordering principle used in stack?
LIFO (Last-in first-out)

28. What does variable **top** signify? (Program 13 & 14)
It points to the topmost element in the stack.
Its value is -1 when stack is empty (Underflow)
If the size of the stack is N, then the value of top will be N-1 when the stack is full (overflow)

29. Mention any two applications of stack in real world.
Stack of plates or books, Batteries in a torch, CDs is in CD rack

30. Mention any two applications of stack in computer science.
Towers of Hanoi, Recursion, Evaluation of expression, Reversing a word, Undo operations in a text editor, Conversion of decimal number to binary

31. Operations on stack
stack() – create a new stack that is empty
push – adds a new item to the top of the stack
pop – removes the top item of the stack
peek – returns the top item of the stack but does not remove it
isEmpty() – tests whether the stack is empty
size() – returns the number of items on the stack

32. What is overflow and underflow condition in a stack?
If the stack is full we cannot insert a new item onto the stack. This is called overflow condition. If the stack is empty we cannot remove an item from the stack. This is called underflow condition.

33. What is a queue?
An ordered collection of items where an item is inserted at one end called the ***rear*** *and an existing item is removed from the other end called the* ***front***

34. What is the ordering principle used in queues?
FIFO (first-in, first-out)

35. Mention the types of queues.
Simple queue, Circular queue, Priority queue and Double ended queue

36. Operations on queue

queue() – creates a new queue that is empty
enqueue – adds a new item to the rear of the queue
dequeue – removes the front item from the queue
isEmpty() – tests to see whether the queue is empty
size() – returns the number of items in the queue

37. Mention any two applications of queue in real world.
Line of students to pay fees, line at a ticket counter, line at bank cash counter

38. Mention any two applications of queue in computer science.
Simulation, Multi-programming platform systems, Scheduling algorithms, Printer server routines, Round-robin technique, Resource-sharing management

39. Can the middle element of a stack/queue be accessed directly?
No, we cannot access the middle element directly

Classes & Objects

1. Define a class.
Template for objects or collection of objects having similar characteristics

2. What is an object?
Basic real world entity with common behaviour and characteristics

3. What is class definition and class declaration?
Class definition – process of naming a class and data variables and interface operations of the class

Class declaration – specifies the representations of objects of the class and the set of operations that can be applied on such objects

4. What are the two types of members referenced in a class?
Data members and member functions

5. What are data members of a class?
The data variables are known as member data of a class, which describe the characteristics of a class

6. What are member functions of a class?
Set of operations that are performed on the data members or set of operations applied on the objects of the class

7. What are access specifiers?
Access specifiers define the scope of data

8. Mention the access specifiers used in a class.
Private, Public and Protected

9. Is it possible to access data outside a class?
Yes, it is possible to access data outside a class if it is a public member

10. Which access specifier is implicitly used in a class?
Private

11. Mention the operator used to access members of a class.
Dot (.) operator is used to access members of a class

12. What is the significance of scope resolution operator (::) with respect to a class?
Scope resolution operator is used to define member functions outside the class

13. Name the data members and member functions in the program.
PS: This can be asked for any of the C++ program. The declaration of data members and member functions in a class needs to be shown

14. What is the formula used to calculate discriminant of a quadratic equation? (Prg 7)
*d=b*b – 4*a*c*

15. Show the line where class definition begins, object is declared and member functions are accessed.
PS: This can be asked for any of the C++ program. The place where class is defined, object is declared or created and member functions are accessed needs to be shown

<u>Function overloading, Inline and friend function</u>

1. What is meant by function overloading?
Function overloading is a feature of C++ that allows us to create multiple functions with the same name but with different parameters or number of parameters.

2. Mention an advantage of function overloading.
   - *Code maintenance is easy*
   - *Code is executed faster*
   - *Easier to understand the flow of information and debug*
   - *Easier interface between programs and real world objects*

3. Name one condition for overloading of functions?
Each function in a set of overloaded functions must have different argument list.

4. What is an inline function?
Inline function is a function for which compiler replaces every function call by the body of the function at compile time

5. How do we define a function to be inline?
By using the keyword inline while defining the function

6. Where are inline functions defined in a program?
Above all the functions or inside a class

7. Mention one advantage of inline function.
   - *Speed of execution of a program increases*
   - *Readability of the program increases*
   - *Very efficient code is generated*

8. Mention one situation when inlining may not work.
   - *If the function definition is too long or too complicated*
   - *If the function is recursive*
   - *If the function has looping constructs*
   - *If the function has a switch or a goto*

9. What is a friend function?
   Friend function is a non-member function of a class that has full access rights to the private and protected members of the class

10. When do we use the keyword friend, while defining the function or declaring it as a friend of a class?
   While declaring the function as a friend of a class

11. Mention any two characteristics of friend function.
   - *Friend function cannot be called using the objects of the class*
   - *They have full access rights to private and protected members if a class*
   - *They are normal external functions that are given special access privileges*
   - *The function is declared using the keyword friend*

Constructors and Destructors

1. What is a constructor?
   A constructor is a special member function that is used in classes to initialize the objects of a class automatically.

2. Write one reason which defines the need to use a constructor.
   To initialize an object of a class automatically without the need to make a separate call to a member function.

3. How do you identify constructor in a program? OR How is the name of a constructor, defined for a class?
   Constructor always has the same name as that of the class name of which they are members.

4. What is the return type of constructors?
   There is no return type for constructors (not even void)

5. What should be the access modifier for constructor declaration?
   Constructors should have either public or protected access.

6. How is the constructor called or invoked?
   A constructor is invoked automatically when objects are created.

7. Mention any one rule for writing a constructor function
   - *Constructor always has the same name as that of the class name of which they are members. This helps the compiler to identify them*
   - *There is no return type for constructors (not even void).*
   - *Defined like any other member function either inside the class or outside*
   - *Should be declared in the public section*
   - *A constructor is invoked automatically when objects are created. They can have default arguments*

- ✸ *A class can have more than one constructor.*

8. Mention the different types of constructors?
Default constructor, Parameterized constructor, Copy constructor

9. What is a default constructor?
A constructor that does not take any arguments is called default constructor. It is also known as zero argument constructor

10. How many default constructors can exist for a class?
A class can have one default constructor

11. What type of constructor is used in program 10?
Parameterized constructor.

12. What is a parameterized constructor?
A constructor that takes one or more arguments is called parameterized constructor.

13. What are the different methods through which constructors are invoked?
Explicit call, Implicit call, Initialization at the time of declaration with = operator.

14. What type of method is used to invoke constructor in program 10?
Implicit call.
An implicit call means the declaration of the object is followed by argument list enclosed in parentheses.

15. How can an object be copied into another?
By using copy constructor.

Inheritance

1. What is inheritance?
Inheritance is the capability of one class to acquire the properties of another class.

2. What is base class?
The class whose properties are inherited by another class. It is also called super class.

3. What is derived class?
The class that inherits the properties from base class. It is also called sub class.

4. Show the derived class header in program 11
class result : public student

5. Mention any one advantage of inheritance.
   - ✸ *Reusing existing code*
   - ✸ *Faster development time*
   - ✸ *Easy to maintain*
   - ✸ *Easy to extend*
   - ✸ *Better memory utilization.*

6. What is visibility mode?

Visibility modes control the access of inherited members within the class.

7. What are the visibility modes that can be used while inheriting a class from another class?
   public, private or protected

8. Identify the base class and derived class in program 11.
   base class – student
   derived class – result

9. What are the different types of inheritance?
   Single inheritance, Multi-level inheritance, Multiple inheritance, Hierarchical inheritance and Hybrid inheritance.

10. What is single / multiple / hierarchical / multi-level / hybrid inheritance?
    Single – If a class is derived from a single base class it is called single inheritance.
    Multilevel – If a class is derived from another derived class it is called multilevel inheritance.
    Multiple – If a class is derived from more than one base class it is known as multiple inheritance.
    Hierarchical – When the properties of one class are inherited by more than one class it is known as Hierarchical inheritance.
    Hybrid – Hybrid inheritance is a combination of both Hierarchical and Multilevel inheritance.

11. What type of inheritance is used in program 11?
    Single Inheritance

<u>Pointers</u>

1. What is a pointer?
   A pointer is a variable that holds the memory address of another variable.

2. Mention any one advantage of pointer.
   - *It is possible to write efficient programs*
   - *Memory is utilized properly*
   - *Dynamic allocation and de-allocation of memory*
   - *Easy to deal with hardware components*
   - *Establishes communication between program and data.*

3. How do you declare a pointer?
   *The general form is data-type *variable_name;*
   *data-type is any valid data type supported by C++ or any user defined type and variable_name is the name of the pointer variable. The presence of * indicates that it is a pointer variable.*
   *Example: int *iptr;*

4. How do we initialize a pointer?
   To initialize a pointer we can assign the address of a variable to a pointer variable.
   int num=25;
   *int *iptr;*
   iptr=&num;
   In the above example, the address of variable num is assigned to pointer variable iptr.

5. What is address of operator?

& is the address of operator. It is a unary operator that returns the memory address of its operand.

6. What is pointer operator or indirection operator or value at address of operator?
   ** is the pointer operator. It is a unary operator that returns the value of the variable located at the address specified by its operand.*

7. What are object pointers?
   Pointers pointing to objects are referred to as object pointers. It is the pointer to an object of an already defined class.

8. How do we access members of a class using an object pointer?
   To access members of a class using object pointer, the arrow operator (->) is used.

9. Identify pointer declaration and initialization in program 12.
   Pointer declaration
   *student *sptr;*

   Pointer initialization
   sptr=&s

<u>Database concepts</u>

1. What is data?
   Data is a collection of facts, figures, statistics, which can be processed to produce meaningful information.

2. What is information?
   Information is processed data with some definite meaning.

3. What are the types of data processing?
   Manual and Electronic data processing.

4. Define database.
   A database is a collection of logically related data organized in a way that data can be easily accessed, managed and updated.

5. Mention the applications of database.
   Banking, Water meter billing, Rail and airlines, Colleges, Credit card transactions, Telecommunications, Finance, Sales, Manufacturing, Human resources.

6. What is a file?
   Basic unit of storage in computer system that is a large collection of related data.

7. What is a relation or a table?
   In relational database, a table or a relation is a collection of data elements organized in terms of rows and columns.

8. What is an entity?
   An entity refers to any real world object that needs to be represented in a database.

9. What is a tuple or record?
   A row of a table that represents a complete information about a particular entity is called as a record or a tuple. It is a collection of related fields.

10. What is a field or an attribute?
    An attribute or a field describes the characteristics of an entity. It is represented by the columns in a table.

11. What is a domain with respect to database?
    Set of all permissible values that can be assigned to an attribute or column in a table.

12. What is DBMS?
    Database Management System is a software that allows creation, definition and manipulation of database.

13. Mention popular DBMS softwares.
    MySQL, Oracle, SQL Server, Sybase, IBM DB2, Microsoft Access.

14. Mention the data types supported by DBMS.
    Integer, Single and double precision floating point numbers, characters, logical data type, strings, Memo data type, Index fields, Currency fields, Date fields, Text fields.

15. What is relational database model?
    Relational database model: Relational model represents the database as a collection of relations or tables of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.

16. Who developed relational model?
    E. F. Codd developed relational model

17. What is primary / candidate/ alternate or secondary / foreign / composite key?
    Candidate key - All those attributes or set of attributes that uniquely identify a tuple are called candidate keys
    Primary key - A primary key is an attribute or set of attributes that uniquely identifies a tuple within a relation.
    Alternate key or Secondary key - Alternate Keys of any table are simply those candidate keys which are not currently selected as the primary key. They are also known as secondary keys
    Composite key - If more than one field together act as a primary key then such a key is called Composite key
    Foreign key - A non-key attribute whose values are derived from the primary key of another table is called as a Foreign Key

<u>SQL</u>

1. Expand and define SQL.
   Structured Query Language is a language of relational database that allows database creation, deletion, modification, retrieval and other operations on database.

2. What are the types of SQL commands?
   DDL (Data Definition Language), DML (Data Manipulation Language), DCL (Data Control Language), TCL (Transaction Control Language)

3. Mention RDBMS packages.
   MySQL, MS Access, Oracle, Sybase, Informix, PostgreSql, MS SQL Server

4. What is DDL?
   Data Definition Language commands are used for creating, modifying, and dropping (deleting) the structure of database objects

5. Mention DDL commands.
   CREATE - to create objects in the database
   ALTER - alters the structure of the database
   DROP - delete objects from the database

6. What is DML?
   Data Manipulation Language commands are used for storing, retrieving, modifying, and deleting data.

7. Mention DML commands.
   According to the original SQL specification the commands are:

   SELECT - retrieve data from the a database
   INSERT - insert data into a table
   UPDATE - updates existing data within a table
   DELETE - deletes all or specific records from a table

8. What is DCL?
   Data Control Language (DCL) is used to control privilege to perform any operation in the database.

9. Mention DCL commands.
   GRANT - gives user's access privileges to database
   REVOKE - withdraw access privileges given with the GRANT command

10. What is TCL?
    Transaction Control (TCL) statements are used to manage the changes made by DML statements. It allows statements to be grouped together into logical transactions.

11. Mention TCL commands.
    COMMIT - permanently save any transaction into database
    SAVEPOINT - identify a point in a transaction to which you can later roll back
    ROLLBACK - restore database to original since the last COMMIT

12. What is DQL?
    Generally Data Query Language is considered to have one command SELECT to retrieve data from the database. If DQL is considered then DML will have only three commands INSERT, UPDATE, DELETE

13. Mention SQL data types.
    int (Number), float (numeric or decimal), datetime (or date and time separately), char, varchar

14. What do you mean by SQL constraints? Mention.
SQL constraints are the set of rules that can be applied on one or more columns on a table.

Constraints are:
NULL – A column can have null value (can be blank)
NOT NULL – A column cannot have null value (there must be some value)
UNIQUE – Column cannot have duplicate value
PRIMARY KEY – Specified column is primary key, cannot have duplicate value, can be used as foreign key in some other table
FOREIGN KEY – The column is already a primary key in some other table. REFERENCES – To specify reference for foreign key to the table having it as primary key (link to the table having this key as primary key)
DEFAULT / SET DEFAULT – Set all values to default value in a column
CHECK – Used to limit the value range that can be placed in a column

15. What is the SQL command used to modify the name of a column?
ALTER command

16. What is the SQL command used to remove a table definition, data and other specifications of a table? OR What is the SQL command used to delete a table?
DROP command.

17. Give the command to display all the details in a table?
SELECT command

18. How do you remove row / rows from a table?
Using DELETE command.

19. How do you display the contents in ascending or descending order based on one or more columns?
Using SELECT command with ORDER BY clause

20. How do you arrange identical data into groups?
Using SELECT command with GROUP BY clause

21. Mention SQL operators.
Arithmetic, Comparison, Logical, Bitwise operators.

22. What is DISTINCT keyword used for?
DISTINCT keyword is used in conjunction with SELECT statement to eliminate all the duplicate records and fetching only unique records.

23. What are group functions? Name them.
Group functions or aggregate functions are built-in functions that operate on groups of rows and return one value for the entire group. These functions are COUNT, MAX, MIN, AVG and SUM.

24. What is a sub query?

A sub query or Inner query or Nested query is a query within another SQL query and embedded within the WHERE clause. A sub query is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

25. What is a database view? How do you create it?
    A view is a virtual table whose data is derived from one or more base tables. It only stores its definition but it does not contain any copy of the data. It can be operated like any other table.

26. Why is join clause used?
    A SQL join clause combines records from two or more tables in a relational database. A JOIN is a means for combining fields from two tables (or more) by using values common to each.

27. What is the use of NULL value? Give example.
    SQL NULL is the term used to represent a missing value. A NULL value in a table is a value in a field that appears to be blank. The NULL value in a column can be searched for in a table using IS NULL in the WHERE clause as below,

    *select * from employee where desig IS NULL;*

Web Designing

1. Define WWW.
   The World Wide Web (abbreviated as WWW or W3, commonly known as the web), is a system of interlinked hypertext documents accessed via the Internet.

2. What is a webpage?
   A webpage is a document or information resource that is suitable for the World Wide Web and can be accessed through a web browser.

3. What is a website?
   A website is a collection of related web pages, including multimedia content, typically identified with a common domain name, and published on at least one web server.

4. What is a homepage?
   A home page or a start page is the initial or main web page of a website or a browser.

5. What is web server?
   Web server can refer to either the hardware (the computer) or the software (the computer application) that helps to deliver Web content that can be accessed through the Internet

6. What is web browser?
   A web browser is a software application for retrieving, presenting, and traversing information resources on the World Wide Web

7. Give an example for a web browser.
   Internet Explorer, Mozilla Firefox, Google Chrome

8. What is URL?
   Uniform Resource Locator (URL) is an internet address or web address that is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it

9. What is HTML?
   Hyper Text Markup Language (HTML) is the language used to create web pages

10. What will be the extension of HTML file?
    .html is the extension of HTML file.

11. Mention the tags related to table.
    <table> - creates a table
    <caption> - represents the title of the table
    <tr> - indicates table row
    <th> - indicates table heading
    <td> - indicates table data

12. What are these attributes?
    cellspacing - Sets the width, in pixels, between individual cells
    cellpadding - Sets the width, in pixels, between the edge of a cell and its content
    rowspan - Allows n number of rows to be combined in a cell
    colspan - Allows n number of columns to be combined in a cell
    nowrap - Used to turn off word wrapping in a cell

13. What is a form in HTML?
    HTML form on a web page allows a user to enter data that is sent to a server for processing.

14. What are the fields and the tags associated with them that can be added in a form?
    Text field - <input type="text"> defines a one-line text input field that allows the user to input text.
    Password field - <input type="password"> defines a password field. The characters in a password field are masked (shown as asterisks or circles)
    Radio button - <input type="radio"> defines a radio button that lets a user to select ONLY ONE of a limited number of choices.
    Check box - <input type="checkbox"> defines a checkbox that lets a user to select ZERO or MORE options of a limited number of choices.
    File upload field - <input type="file"> defines a file-select field and a "Browse..." button (for file uploads)
    Hidden field - <input type="”hidden”> is useful for situations when you might want to gather data not supplied by the person filling out the form.
    Button - <input type="“button”> creates a clickable button, that activates a JavaScript when it is clicked
    Submit button - <input type="”submit”> - Defines a button that is clicked to submit a form.
    Image button - <input type="“image”> allows to substitute an image for the standard submit button
    Reset button - <input type="“reset”> defines a button on a form that will return all fields to their default values generally clearing all the input in the form
    Text area - <textarea> element defines a multi-line input field (a text area).
    Drop down list or list box - The <select> element is used to create a drop-down list or list box. The <option> tags inside the <select> element define the available options in the list. When size attribute of <select> tag has value 1, it behaves like a drop down list. If size is greater than 1, it behaves as a list box

15. What is Web Scripting?

Process of creating and embedding scripts in a web page. Types are Client side scripts (supports interaction within a web page) and server side scripts (supports execution at the server end)

16. What is:

Searching – Searching the Internet means looking for something identifiable

Surfing – Surfing the Internet means hopping from one computer to another in order to look for information which may be available on more than one server in the Internet and which are linked by the WWW

Telnet – It is a user command and an underlying TCP/IP protocol for accessing remote computers.

Search Engine - Internet search engines are special sites on the Web that are designed to help people find information stored on other sites. E.g.: google.com, altavista.com, yahoo.com, bing

17. What is Web hosting?

Web hosting is a service that allows organizations and individuals to post a website or web page onto the Internet.

18. Mention the types of Web hosting?

Free hosting, Share hosting, Dedicated hosting, Co-location hosting

19. Mention any one advantage of HTML?

- *Easy and quick to develop*
- *Simple to edit, only requires a text editor*
- *It is widely used and every browser supports HTML language*
- *HTML is designed with a feature of interaction between user and web page, which makes it effective*
- *Follows client/server architecture and reduces overhead on server as the document is processed by client*

20. Mention any one disadvantage of HTML?

- *Users have free access to look at HTML source code*
- *No provision for security*
- *It can create only static and plain pages so if we need dynamic pages then HTML is not useful.*
- *Need to write lot of code for making simple webpage*
- *Sometimes, the structuring of HTML documents is hard to grasp*

21. Mention any one advantage of Web Designing.

- *It increases the popularity of a company*
- *It helps a company to increase product knowledge and communication between company and customers*
- *It provides all information about products, advertising as well as development activities*
- *Potential to reach a wider audience*
- *Business information and details about products and services can be accessed by anyone no matter where they are and what time it is*
- *Website can be kept up to date to be relevant to the business and encourage more customers*