HIGHER SECONDARY LAB MANUAL

Modified for the Academic Year 2023 Onwards

Solved Problems of Practical Evaluation Questions - Computer Science

Programming in C++, Web development Using HTML and SQL

Detailed procedure for program execution and debugging.

Prepared By:

Dr. Nishad Abdulkareem
HSST Computer Science
Govt. Tribal HSS, Sholayoor, Palakkad

Preface

The programs showcased in this book align with the Guidelines for laboratory work and the Practical Evaluation of Computer Science - 2023 (Kerala Higher Secondary Education). At the end of each section, you'll find comprehensive instructions for coding and executing C++, HTML, and SQL codes.

Since the programs created here are intended for educational purposes, we use straightforward logical constructs, and certain necessary data validations are intentionally omitted for the sake of simplicity.

The document is formatted using LaTeX, the de facto standard for creating and publishing scientific documents.

Please share any comments with the author at an.nishad@gmail.com.

[1] You can access **the detailed demonstrations for all the listed codes in the playlist** Lab program illustration

[1]Document Version : 3.0

Contents

1 Programming in C++

1.1 Roots of quadratic equation

PROBLEM

Input the three coefficient of quadratic equation and find the roots

CODE


```
#include<iostream>
#include<cmath>
using namespace std;
int main() {
    double a, b, c;
    double root1,root2, realPart, imgPart;
    cout <<"Enter the coefficients of the quadratic equation
   (a, b, c):";
    cin >> a >> b >> c;
    double disc = b * b - 4 * a * c;
    if(disc > 0) {
      root1 = (-b + sqrt(disc)) / (2 * a);
        root2 = (-b - sqrt(disc)) / (2 * a);
         cout << "Real and Imaginary roots\n";
```

```
14          cout << "Root 1: " << root1 <<"\n";
15          cout << "Root 2: " << root2 <<"\n";
16      } else if (disc == 0) {
17          root1 = -b / (2 * a);
18          cout << "Two Roots are equal.\n";
19          cout << "Root: " << root1;
20      } else {
21         realPart = -b / (2 * a);
22         imgPart = sqrt(-disc) / (2 * a);
23         cout<<"Complex Roots \n";
24         cout << "Root 1: "<< realPart << " + " << imgPart << "
    i\n";
25         cout << "Root 2: "<< realPart << " - " << imgPart << "
    i\n";
26      }
27      return 0;
28  }
```

Listing 1.1: quadratic.cpp

OUTPUT

```
Enter the coefficients of the quadratic equation (a, b, c):2
5
7
Complex Roots
Root 1: -1.25 + 1.39194i
Root 2: -1.25 - 1.39194i
```

```
Enter the coefficients of the quadratic equation (a, b, c):-2 5 7
Real and Imaginary roots
Root 1: -1
Root 2: 3.5
```

```
Enter the coefficients of the quadratic equation (a, b, c):1 2 1
Two Roots are equal.
Root: -1
```

Figure 1.1: Roots of quadratic equation

1.2 Display word corresponding to a digit

PROBLEM

Input a digit and display the same in word. (Zero for 0, One for 1 etc... Nine for 9)

CODE

```
#include<iostream>
using namespace std;
int main(){
    int num;
    cout<<"Input the digit to display in words:";
    cin>>num;
    switch(num)
    {
      case 0:
        cout<<"Zero"; break;
      case 1:
        cout<<"One"; break;
      case 2:
        cout<<"Two";break;
      case 3:
        cout<<"Three";break;
      case 4:
        cout<<"Four";break;
      case 5:
        cout<<"Five";break;
      case 6:
        cout<<"Six";break;
      case 7:
```

```
24          cout<<"Seven";break;
25        case 8:
26          cout<<"Eight";break;
27        case 9:
28          cout<<"Nine";break;
29        default:
30          cout<<"Invalid input";
31      }
32    return 0;
33  }
```

Listing 1.2: numtoword.cpp

OUTPUT

```
Input the digit to display in words:5
Five
```

Figure 1.2: Display word corresponding to a digit

1.3 Sum of n natural numbers

PROBLEM

Find the sum of n natural numbers without using a formula

CODE

```
1 #include<iostream>
2 using namespace std;
3 int main() {
```

```
    int n;
    cout<< "Enter a positive integer (n): ";
    cin >> n;
    int sum = 0;
    for (int i = 1; i <= n; i++) {
        sum += i;
      }
    cout <<"The sum of the first " <<n<< " natural numbers is
   : " << sum;
    return 0;
}
```

Listing 1.3: sumofnatural.cpp

OUTPUT

```
Enter a positive integer (n): 11
The sum of the first 11 natural numbers is: 66
```

Figure 1.3: Sum of n natural numbers

1.4 Check integer for palindrome

PROBLEM

Input an integer number and check whether it is a palindrome or not

CODE

```
#include <iostream>
```

```cpp
using namespace std;
int main()
{
     int n, num, digit, rev = 0;
     cout << "Enter a positive number: ";
     cin >> num;
     n = num;
     while (num > 0)
     {
          digit = num % 10;
          rev = (rev * 10) + digit;
          num = num / 10;
     }
     cout << "The reverse of the number is: " << rev;
     if (n == rev)
          cout << " \n The number is a palindrome.";
     else
          cout << " \n The number is not a palindrome.";
    return 0;
}
```

Listing 1.4: ispalindrome.cpp

```
Enter a positive number: 1536
 The reverse of the number is: 6351
 The number is not a palindrome.
```

```
Enter a positive number: 9559
 The reverse of the number is: 9559
 The number is a palindrome.
```

Figure 1.4: Check if the given integer is Palindrome or not

OUTPUT

1.5 Check integer for prime

PROBLEM

Input an integer number and check whether it is a prime or not

CODE

```
1  #include <iostream>
2  using namespace std;
3  int main(){
4   int num;
5   bool isPrime=true;
6   cout <<"Provide a positive number: ";
7   cin >> num;
8   if(num<=1)
9    isPrime=false;
10  else{
11   for(int i = 2; i <= num/2; i++) {
12     if (num % i == 0) {
13       isPrime = false;
```

```
14      }
15    }}
16   if(isPrime)
17      cout<<"Number is prime";
18   else
19      cout<<"Number is not prime";
20   return 0;
21  }
```

Listing 1.5: isprime.cpp

OUTPUT

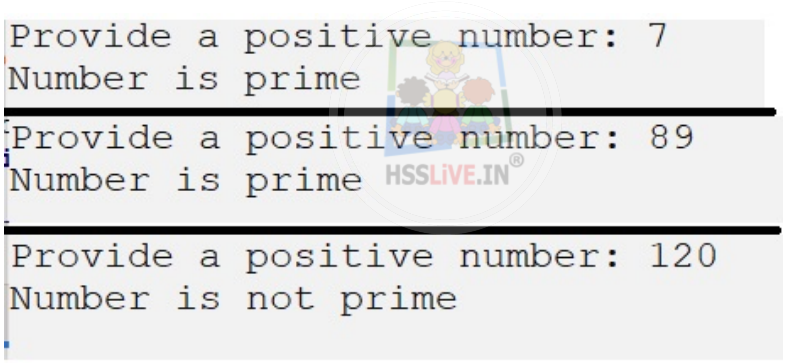


Figure 1.5: Check if the given integer is Prime or not

1.6 Sorting of array

PROBLEM

Create an array to store the heights of some students and sort the values

CODE

```cpp
#include <iostream>
using namespace std;
int main() {
    int heights[20], i, j,n, temp;
    cout<< "Enter the number of students:\n";
    cin>>n;
  cout<< "Enter the heights of  students:\n";
    for (i = 0; i < n; i++) {
        cout << "Student " << (i + 1) << ": ";
        cin >> heights[i];
    }
    for(i=0; i<n; i++)
    {
       for(j=i+1; j<n; j++) {
    if(heights[i]>heights[j])
    {
      temp = heights[i];
      heights[i] = heights[j];
      heights[j] = temp;
    }
        }
    }
    cout << "The Sorted Heights \n";
      for (i = 0; i < n; i++) {
        cout << heights[i]<< "\n ";
          }
    return 0;
}
```

Listing 1.6: arraysort.cpp

OUTPUT

```
Enter the number of students:
10
Enter the heights of  students:
Student 1: 123
Student 2: 100
Student 3: 186
Student 4: 149
Student 5: 146
Student 6: 153
Student 7: 153
Student 8: 148
Student 9: 169
Student 10: 163
The Sorted Hieights
100
 123
 146
 148
 149
 153
 153
 163
 169
 186
```

```
Enter the number of students:
6
Enter the heights of  students:
Student 1: 150
Student 2: 165
Student 3: 100
Student 4: 156
Student 5: 103
Student 6: 134
The Sorted Hieights
100
 103
 134
 150
 156
 165
```

Figure 1.6: Array sorting using bubble sort

1.7 Compute length of a string

PROBLEM

Find the length of a string without using *strlen*() function

CODE


```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    char str[50];
  int i,length=0;
  cout<<"Enter the string:";
  gets(str);
  for(i=0;str[i]!='\0';++i)
    length++;
  cout<<"Length of the string is:"<<length;
  return 0;
}
```

Listing 1.7: stringlength.cpp

```
Enter the string:apple
Length of the string is:5
```

```
Enter the string:I never dreamed about success. I worked for it
Length of the string is:46
```

```
Enter the string:15August 1947
Length of the string is:13
```

Figure 1.7: Counting the length of a given string

1.8 Compute nCr using a user-defined function

PROBLEM

Define a function to find the factorial of a number. Using this function find the value of nCr

CODE

```
#include <iostream>
using namespace std;
long fact(int n) {
    long result = 1;
    for (int i = 1; i <= n; ++i) {
        result = result*i;
    }
    return result;
}
int main() {
```

```
11    int n, r,ncr;
12    cout<<"Enter the value of n and r (positive integer)";
13    cin >>n>>r;
14   if((n>=r)&& (r>=0)){
15     ncr = fact(n) / (fact(r) * fact(n - r));
16     cout<< "nCr is: " << ncr;
17   }
18    else
19       cout<<"n and r must be non-negative integers and n > r.
     " ;
20    return 0;
21 }
```

Listing 1.8: ncr.cpp


```
Enter the value of n and r (positive integer)5 2
nCr is: 10
```

```
Enter the value of n and r (positive integer)8 2
nCr is: 28
```

```
Enter the value of n and r (positive integer)2 8
n and r must be non-negative integers and n > r.
```

```
Enter the value of n and r (positive integer)8 0
nCr is: 1
```

```
Enter the value of n and r (positive integer)0 8
n and r must be non-negative integers and n > r.
```

Figure 1.8: nCr for different n and r

1.9 Student record management using *structure*

PROBLEM

Create a structure to represent admission number, name and marks given for CE, PE of a subject. Input the details of student and display admission number, name and total marks obtained

CODE

```
#include <iostream>
#include <cstdio>
using namespace std;
struct student{
  int admNo;
  char name[25];
  int ceMark;
  int peMark;
  int totalMark;
};
int main() {
    student s;
    cout<<"Enter the student details:\n";
    cout<<"\n Admission Number: ";
    cin>>s.admNo;
    cout<<"\n Name : ";
    cin>>ws; //
    gets(s.name);
    cout<<"\n CE Mark : ";
    cin>>s.ceMark;
    cout<<"\n PE Mark : ";
    cin>>s.peMark;
```

```
23      s.totalMark = s.peMark + s.ceMark;
24      cout<<"\n The Student Details \n";
25      cout<<"\nName : "<<s.name;
26      cout<<"\nAdmission No. : "<<s.admNo;
27      cout<<"\nCE Mark : "<<s.ceMark;
28      cout<<"\nPE Mark : "<<s.peMark;
29      cout<<"\nTotal Mark : "<<s.totalMark;
30      return 0;
31  }
```

Listing 1.9: studentstruct.cpp

1 2

OUTPUT

[1] *cin >> ws* tells the compiler to ignore buffer and also to discard all the white spaces before the actual content of string or character array

[2] To resolve the issue where some IDEs may ignore the *gets* command, follow these steps. Click on "Build Commands" in the "Set Build Commands" section. In the 'Compile' box and 'Build' box, append the command -std=c++11.

```
Enter the student details:

 Admission Number: 1234

 Name : Sibi Jacob

 CE Mark : 19

 PE Mark : 48

 The Student Details

Name : Sibi Jacob
Admission No. : 1234
CE Mark : 19
PE Mark : 48
Total Mark : 67
```

Figure 1.9: Data populated from the student structure

1.10 Integer swap operation using a user-defined function with pointer arguments

PROBLEM

Input two numbers swap them by defining a function with pointer arguments

CODE

```
#include<iostream>
using namespace std;
void swap(int *x,int *y)
{
  int t;
  t=*x;
  *x=*y;
```

```
8   *y=t;
9 }
10 int main( )
11 {
12   int a,b;
13   cout<<"Enter the values of a and b";
14   cin>>a>>b;
15   cout<<"\nBefore Swap \n";
16   cout<<"a="<<a<<"\t b="<<b;
17   swap(&a,&b);
18   cout<<"\nAfter Swap \n";
19   cout<<"a="<<a<<"\t b="<<b;
20 }
```

Listing 1.10: swappointer.cpp

OUTPUT

```
Enter the values of a and b 25 123

Before Swap
a=25     b=123
After Swap
a=123    b=25
```

Figure 1.10: Swapping of two numbers

1.11 Check whether the given number is positive, negative, or zero

PROBLEM

Input a number and check if it is positive negative or zero

CODE


```
#include<iostream>
using namespace std;
int main(){
  int num;
  cout<<"Enter the integer to check :";
  cin>>num;
  if(num>0)
    cout<<"\n The given number is Positive";
  else if(num<0)
    cout<<"\n The given number is Negative";
  else
    cout<<"\n The given number is Zero";
  return 0; }
```

Listing 1.11: numcheck.cpp

```
Enter the integer to check :0
 The given number is Zero
Enter the integer to check : -158
 The given number is Negative
Enter the integer to check :3256
 The given number is Positive
```

Figure 1.11: number check

1.12 Check whether the given number is odd or even

PROBLEM

Input a number and check if it is odd or even

CODE

```
#include<iostream>
using namespace std;
int main()
{
  int num;
  cout<<"Enter a number to check-odd or even :";
  cin>>num;
  if(num%2==0)
    cout<<"The given number is even";
  else
    cout<<"The Given number is odd";
  return 0;
```

```
}
```

Listing 1.12: oddeven.cpp

OUTPUT

```
Enter a number to check-odd or even :12
The given number is even
Enter a number to check-odd or even :11
The Given number is odd
```

Figure 1.12: odd even

1.13 Sum of the squares of the first N natural numbers

PROBLEM

Find the sum of squares of integers up to a specified limit by inputting a number.

CODE

```
#include<iostream>
using namespace std;
int main()
{
  int num, sum=0, i;
  cout<<"Enter the limit :";
  cin>>num;
  for(i=1;i<=num;i++)
   sum=sum+i*i;
```

```
10    cout<<"Sum of square:"<<sum;
11    return 0;
12 }
```

Listing 1.13: sumsquare.cpp

OUTPUT

```
Enter the limit :2
Sum of square:5
```

```
Enter the limit :10
Sum of square:385
```

```
Enter the limit :16
Sum of square:1496
```


Figure 1.13: Sum of Squares of integer

2 Developing HTML documents

2.1 Kerala tourism web page

PROBLEM

Design a simple and attractive web page for Kerala Tourism. It should contain features like back ground colour / image headings, text formatting and font tags, images etc.

CODE

```
<html>
  <head>
    <head>
    <title> Kerala Tourism</title>
  </head>
  <body background="mountains.jpeg" bgcolor ="Green">
    <h1 align="center">DEPARTMENT OF TOURISM</h1>
    <h2 align="center">Kerala State</h2>
    <b>Kerala</b>, in the south-western part of India,
         is a highly desirable tourist destination in Asia,
         often called
        <font size="10" color="blue " face="Cambria">
        God's Own Country.</font></br>
```

```
14    It was recognized by National Geographic
15      Traveller as one of the world's 50 lifetime destinations
16      and one of the thirteen <em>paradises on Earth.</em>
17    </body>
18  </html>
```

Listing 2.1: tourism.html

Figure 2.1: Screenshot of website

2.2 Hyperlinking to Various Web Pages

PROBLEM

Design a simple web page about your school. Create another web page named address.html containing the school address. Give links from school page to address.html and reverse

CODE

school.html

```
<html>
<head>
    <title>My School</title>
</head>
<body background="school.jpeg" >
  <h1 align="CENTER" color="blue"> Got. Tribal HSS Sholayoor
   </h1>
  <br>Our school, located in the picturesque hilly
  region of Sholayoor within Palakkad District, is nestled
  in the captivating
  Attappadi mountain  a r e a a  renowned hill station in Kerala
   .
  The school proudly offers four distinct batches of
   Higher Secondary Courses, namely Biology Science,
  Home Science, Commerce, and Sociology.
  <br>
  From first standard to twelve about 800 students
  are studying here.
  <BR>With a student body ranging from the first standard
  to the twelfth,we currently educate approximately 800
   students.
   Furthermore, our school is supported by a dedicated team
   of
   about 35 staff members who work diligently
   to provide quality education.
   <br>
   <a href="address.html"> Click here for our school address<
   /a>
</body>
```

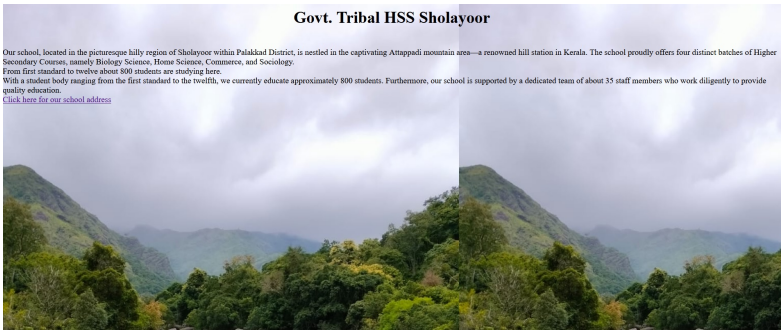
```
25 </html>
```

Listing 2.2: school.html

address.html

```
1  <html>
2      <head>
3          <title>Address</title>
4      </head>
5  <body>
6      <h2 align="center"> Govt Tribal HSS </h2>
7      <address>
8          Govt Tribal HSS Sholayoor<BR>
9          Sholayoor Post<BR>
10         Attappadi , Mannarkkad<br>
11         Palakkad Dist<br>
12     </address>
13     <a href="school.html"> back to school</a>
14 </body>
15 </html>
```

Listing 2.3: address.html



(a) Screenshot of school homepage

(b) Screenshot of school address page

2.3 Unordered List Tags

PROBLEM

Design a Web page as shown in 2.3 using appropriate List Tags

CODE

```
<html>
  <head>
    <title>Higher Education Instituitions</title>
  </head>
```

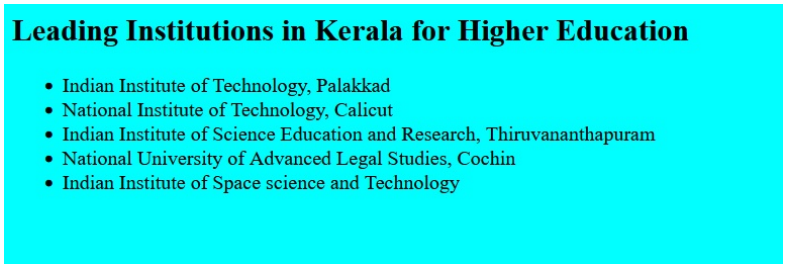


Figure 2.3: Unordered list

```
5   <body bgcolor ="Cyan">
6     <h2>Leading Institutions in Kerala for Higher Education</
    h2>
7     <ul>
8       <li>Indian Institute of Technology, Palakkad</li>
9       <li>National Institute of Technology, Calicut</li>
10      <li>Indian Institute of Science Education and Research,
    Thiruvananthapuram</li>
11      <li>National University of Advanced Legal Studies,
    Cochin</li>
12      <li>Indian Institute of Space science and Technology</
    li>
13    </ul>
14  </body>
15 </html>
```

Listing 2.4: highereducate.html

Terrestrial Planets(Source NASA)

Planet	Day Length (In Earth hour)	Year Length (In Earth days)
Mercury	1408	88
Venus	5832	224.7
Earth	24	365.26

Figure 2.4: Terrestrial Planets listed in a Table

2.4 Table of terrestrial planets

PROBLEM

Design a Web page containing a table as shown in Figure 2.4

CODE


```
<html>
<head>
<title>Terrestrial Planets</title>
</head>
<body>
  <table border="1">
  <caption><b>Terrestrial Planets</b>(Source NASA)</caption>
    <tr>
      <th>Planet </th>
      <th> Day Length <br> (In Earth hour) </th>
      <th> Year Length <br> (In Earth days) </th>
    </tr>
    <tr>
      <td>Mercury</td>
```

```
      <td>1408</td>
      <td>88</td>
    </tr>
    <tr>
      <td >Venus</td>
      <td>5832</td>
      <td>224.7</td>
    </tr>
    <tr>
      <td >Earth</td>
      <td>24</td>
      <td>365.26</td>
    </tr>
  </table>
</body>
</html>
```


Listing 2.5: terrestrial.html

2.5 Client login page

PROBLEM

Design a web page as shown in Figure 2.5

CODE

```
<HTML>
  <HEAD><TITLE>LOGIN</TITLE></HEAD>
<BODY>
<FORM>
<TABLE>
```

Figure 2.5: Client login page

```
6       <TR><TH>Client Login </TH></TR>
7        <TR><TD>User Name </TD><TD><INPUT Type="Text"></TD></TR
    >
8        <TR><TD>Password</TD><TD><INPUT Type="Password"></TD></
    TR>
9        <TR>
10        <TD><INPUT Type="Submit" value="Submit"></TD>
11        <TD> <INPUT Type="Reset" value="Clear"></TD>
12      </TR>
13 </TABLE>
14 </FORM>
15 </BODY>
16 </HTML>
```

Listing 2.6: login.html

2.6 Case converter using JavaScript

PROBLEM

A web page should contain one text box for entering a text. There should be two buttons labelled to Upper Case and To Lower Case. on clicking each button, the content in the text box should be converted to upper case or lower case accordingly . Write required java script for this operations

CODE

```
<html>
<head>
  <title>Case Converter</title>
  <script language="javascript">
    // Function to convert text to uppercase
  function convertToUpperCase() {
    var textInput = document.myForm.textInput.value;
    document.myForm.textInput.value = textInput.toUpperCase()
   ;
   }
  // Function to convert text to lowercase
  function convertToLowerCase() {
    var textInput = document.myForm.textInput.value;
    document.myForm.textInput.value = textInput.toLowerCase()
   ;
  }
    </script>
</head>
<body bgcolor="green">
```

```
18 <h2>lower case : the celebration of equality and simplicity
      in language and design</h2>
19 <form name="myForm">
20        <table>
21       <tr><td>Provide a text</td>
22         <td><input type="text" name ="textInput"></td>
23       </tr>
24       <tr>
25         <td><input type="button"  onclick="convertToUpperCase
      ()" value="To Upper Case"></td>
26         <td><input type="button"  onclick="convertToLowerCase
      ()" value="To Lower Case"></td>
27       </tr>
28     </table>
29 </form>
30 </body>
31 </html>
```


Listing 2.7: casechangescript.html

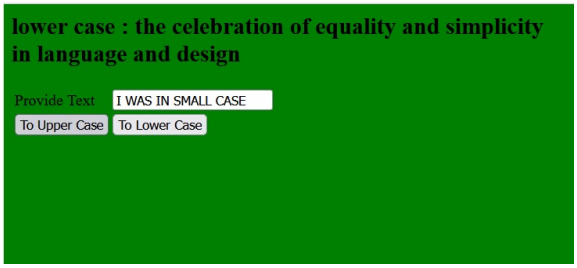


Figure 2.6: Case converter

3 SQL

3.1 Creating a *student* table and perform fundamental operations

PROBLEM

Create a table Student with the following fields and insert at least 5 records into the table except for the column Total.

Roll_Number Integer Primary key,Name Varchar (25),

Batch Varchar (15), Mark1 Integer, Mark2 Integer,

Mark3 Integer,Total Integer

1. Update the column Total with the sum of Mark1, Mark2 and Mark3.
2. List the details of students in Commerce batch.
3. Display the name and total marks of students who are failed (Total less than 90).
4. Display the name of students in alphabetical order and in batch based

SOLUTION

Creation of student table

```
CREATE TABLE student(
  roll_number INT NOT NULL  PRIMARY KEY,
  stud_name VARCHAR(25), batch VARCHAR(25),
  mark1 INT , mark2 INT, mark3 INT , total INT
);
```

Listing 3.1: Query: Create *student* table

Inserting values into the student table

```
insert into student values
    (1,'Alan Turing', 'Science',80, 70, 55,null);
insert into student values
    (2,'Ada Lovelace', 'Science',30, 40, 15,null );
insert into student values
    (3,'Adam Smith', 'Commerce', 70, 60, 40,null);
insert into student values
    (4,'Amartya Sen', 'Commerce',40, 80, 50,null);
insert into student values
    (5,'Ibn Rushd', 'Science', 90, 94, 35,null);
insert into student values
    (6,'Irfan Habeeb', 'Humanities', 80, 76, 69,null);
insert into student values
    (7,'Friedman', 'Commerce', 60, 60, 50,null);
insert into student values
    (8,'Robinson', 'Commerce', 50, 70, 50,null);
```

Listing 3.2: Query:Multiple queries to insert records in to the *student* table.

Fetching complete records from the *student* table

```
SELECT * FROM student;
```

roll_number	stud_name	batch	mark1	mark2	mark3	total
1	Alan Turing	Science	80	70	55	NULL
2	Ada Lovelace	Science	30	40	15	NULL
3	Adam Smith	Commerce	70	60	40	NULL
4	Amartya Sen	Commerce	40	80	50	NULL
5	Ibn Rushd	Science	90	94	35	NULL
6	Irfan Habeeb	Humanities	80	76	69	NULL
7	Friedman	Commerce	60	60	50	NULL
8	Robinson	Commerce	50	70	50	NULL

Figure 3.1: Retrieving all records from the table

```
Answer 1: UPDATE student SET total=mark1+mark2+mark3;
Answer 2: SELECT * FROM student WHERE batch= 'Commerce';
```

roll_number	stud_name	batch	mark1	mark2	mark3	total
3	Adam Smith	Commerce	70	60	40	170
4	Amartya Sen	Commerce	40	80	50	170
7	Friedman	Commerce	60	60	50	170
8	Robinson	Commerce	50	70	50	170

Figure 3.2: Complete student details of *Commerce* batch

```
Answer 3. select stud_name,total from student where total<90;
```

Refer Figure 3.3 for the result

stud_name	total
Ada Lovelace	85

Figure 3.3: Result of select query total <90

```
Answer 4: select stud_name, batch from student order by batch
    , stud_name;
```

stud_name	batch
Adam Smith	Commerce
Amartya Sen	Commerce
Friedman	Commerce
Robinson	Commerce
Irfan Habeeb	Humanities
Ada Lovelace	Science
Alan Turing	Science
Ibn Rushd	Science

Figure 3.4: Result of select - order by query

3.2 Creating an *employee* table and perform fundamental operations

PROBLEM

Create a table employee with the following fields and insert at least 5 records into the table except the column Gross_pay and DA.

Emp_code Integer Primary key, Emp_name Varchar (20),
Designation Varchar (25), Department Varchar (25), Basic Decimal (10,2)
DA Decimal (10,2), Gross_pay Decimal (10,2)

1. Update DA with 75% of Basic.
2. Display the details of employees in Purchase, Sales and HR departments.
3. Update the Gross_pay with the sum of Basic and DA.
4. Display the details of employee with gross pay below 20000

SOLUTION

Creation of employee table

```
    CREATE TABLE employee(Emp_code Int PRIMARY KEY,
    Emp_name VARCHAR(20),Designation VARCHAR(25),
    Department VARCHAR(25),Basic DEC(10,2),
    DA DEC(10,2),Gross_pay DEC(10,2));
```

Listing 3.3: Create Query

Inserting Values to the *employee* table

```
INSERT INTO employee VALUES
    (100,'Arjun','Trainees ','Sales',16000.00,NULL,NULL),
    (101,'Maya','Trainees','Purchase',9000.00,NULL,NULL),
    (102,'Rahul','Trainees','Purchase',10000.00,NULL,NULL),
    (103,'Ibrahim','Supervisors ','Purchase',18000.00,NULL,
   NULL),
    (104,'Abdulla','Supervisors ','Sales',21000.00,NULL,NULL)
   ,
    (105,'Vivek','Managers ','HR',32000.00,NULL,NULL),
   (106,'George','Managers ','Sales',45000.00,NULL,NULL),
   (107,'Meera','Trainees','HR',11000.00,NULL,NULL);
   (108,'Tiji','Trainees','IT',12000.00,NULL,NULL);
```

Listing 3.4: Query :Insert multiple record in to the employee table using a single query

```
Answer 1: UPDATE employee SET DA=Basic*75/100;
Answer 2: SELECT * FROM employee WHERE Department IN ('
   Purchase', 'Sales','HR');
```

Emp_code	Emp_name	Designation	Department	Basic	DA	Gross_pay
100	Arjun	Trainees	Sales	16000.00	12000.00	NULL
101	Maya	Trainees	Purchase	9000.00	6750.00	NULL
102	Rahul	Trainees	Purchase	10000.00	7500.00	NULL
103	Ibrahim	Supervisors	Purchase	18000.00	13500.00	NULL
104	Abdulla	Supervisors	Sales	21000.00	15750.00	NULL
105	Vivek	Managers	HR	32000.00	24000.00	NULL
106	George	Managers	Sales	45000.00	33750.00	NULL
107	Meera	Trainees	HR	11000.00	8250.00	NULL

Figure 3.5: Result of select query from employee table

```
Answer 3: UPDATE employee SET Gross_pay=Basic+DA;
Answer 4: SELECT * FROM employee WHERE Gross_pay < 20000;
```

Emp_code	Emp_name	Designation	Department	Basic	DA	Gross_pay
101	Maya	Trainees	Purchase	9000.00	6750.00	15750.00
102	Rahul	Trainees	Purchase	10000.00	7500.00	17500.00
107	Meera	Trainees	HR	11000.00	8250.00	19250.00

Figure 3.6: Resultant table of select query from the employee table with Gross_pay <20000

3.3 Create a *stock* table and perform fundamental database operations

PROBLEM

Create a table Stock, which stores daily sales of items in a shop, with the following fields and insert at least 5 records into the table.

Item_code Integer Primary key, Item_name Varchar (20)

Manufacturer_Code Varchar (5), Qty Integer, Unit_Price Decimal (10,2)

1. Display the item_name with stock zero.
2. Display the number of items manufactured by the same manufacturer.

3. Display the highest price and lowest quantity in the stock
4. Increase the unit price of all items by 10%

SOLUTION

Creation of stock table

```
CREATE TABLE stock(Item_code INTEGER PRIMARY KEY,
Item_name VARCHAR(20),Manufacturer_Code VARCHAR(5),
Qty INT,Unit_Price DECIMAL(10,2));
```

Listing 3.5: Query: Creation of *stock* table

Inserting values to the stock table

```
INSERT INTO stock (‘Item_code‘, ‘Item_name‘, ‘
   Manufacturer_Code‘,
‘Qty‘, ‘Unit_Price‘) VALUES (’1001’, ’Smart Phone’, ’Samsung’
   , ’23’, ’36000’);
INSERT INTO stock (‘Item_code‘, ‘Item_name‘, ‘
   Manufacturer_Code‘,
‘Qty‘, ‘Unit_Price‘)
VALUES (’1002’, ’Smart Watch’, ’Samsung ’, ’16’, ’19000’);
INSERT INTO stock (‘Item_code‘, ‘Item_name‘, ‘
   Manufacturer_Code‘,
‘Qty‘, ‘Unit_Price‘)
VALUES (’1003’, ’Washing Machine’, ’LG’, ’21’, ’36000’);
INSERT INTO ‘stock (‘Item_code‘, ‘Item_name‘, ‘
   Manufacturer_Code‘,
‘Qty‘, ‘Unit_Price‘)
VALUES (’1004’, ’Juicer’, ’Butterfly’, ’5’, ’7000’);
```

```
INSERT INTO stock (‘Item_code‘, ‘Item_name‘, ‘
   Manufacturer_Code‘,
‘Qty‘, ‘Unit_Price‘)
VALUES (’1005’, ’Toaster’, ’Philips’, ’0’, ’3000’);
INSERT INTO stock (‘Item_code‘, ‘Item_name‘, ‘
   Manufacturer_Code‘,
‘Qty‘, ‘Unit_Price‘)
VALUES (’1006’, ’Air Purifier’, ’Philips’, ’3’, ’9500’);
INSERT INTO stock (‘Item_code‘, ‘Item_name‘, ‘
   Manufacturer_Code‘,
‘Qty‘, ‘Unit_Price‘)
VALUES (’1007’, ’Coffee Maker’, ’Havels’, ’0’, ’16000’);
```

Listing 3.6: Query: Multiple queries to insert various records into the *stock* table.

```
Answer 1: SELECT * FROM stock WHERE Qty=0;
```

Item_code	Item_name	Manufacturer_Code	Qty	Unit_Price
1005	Toaster	Philips	0	3000.00
1007	Coffee Maker	Havels	0	16000.00

Figure 3.7: Resultant table of select query from the stock table with stock = 0

```
Answer 2:
SELECT  Manufacturer_Code, COUNT(Item_code) AS No_Products
    FROM stock
    GROUP BY Manufacturer_Code;
```

Listing 3.7: Query: Selecting items manufactured by same manufacturer

```
Answer 3:
    SELECT MAX(Unit_price) AS Highest_price ,
    MIN(Qty) AS  Min_Qty FROM stock;
```

Listing 3.8: Query: Selecting higherst price and lowest stock

Manufacturer_Code	No_Products
Samsung	2
LG	1
Butterfly	1
Philips	2
Havels	1

Figure 3.8: The quantity of items produced by same manufacturer from the stock database

Highest_price	Min_Qty
36000.00	0

Figure 3.9: The table resulting from the SELECT query


```
Answer 4:
UPDATE stock SET Unit_Price = Unit_Price + Unit_Price*.1;
```

Listing 3.9: Query: Updating unit price

Item_code	Item_name	Manufacturer_Code	Qty	Unit_Price
1001	Smart Phone	Samsung	23	36000.00
1002	Smart Watch	Samsung	16	19000.00
1003	Washing Machine	LG	21	36000.00
1004	Juicer	Butterfly	5	7000.00
1005	Toaster	Philips	0	3000.00
1006	Air Purifier	Philips	3	9500.00
1007	Coffee Maker	Havels	0	16000.00

Figure 3.10: Before executing the Update Query

Item_code	Item_name	Manufacturer_Code	Qty	Unit_Price
1001	Smart Phone	Samsung	23	39600.00
1002	Smart Watch	Samsung	16	20900.00
1003	Washing Machine	LG	21	39600.00
1004	Juicer	Butterfly	5	7700.00
1005	Toaster	Philips	0	3300.00
1006	Air Purifier	Philips	3	10450.00
1007	Coffee Maker	Havels	0	17600.00

Figure 3.11: After executing the Update Query

3.4 Create a *bank* table and perform fundamental database operations

PROBLEM

Create a table bank with the following fields and insert at least 5 records into the table.

Acc_Number Integer Primary key, Acc_Name Varchar (20),

Branch_Name Varchar (25), Acc_Type Varchar (10) , Amount Decimal (10,2)

1. Display the account details account in Kodungallur Branch
2. Change the branch name Trivandrum to Thiruvananthapuram.
3. Display the details of customers in Thiruvananthapuram, Eranakulam and Kozhikkode
4. List the details of customers in Trissur branch having a minimum balance of Rs. 5000

SOLUTION

Creation of *bank* table

```
CREATE TABLE bank (
  Acc_No INT NOT NULL PRIMARY KEY , Acc_Name‘ VARCHAR(20),
  Branch_Name VARCHAR(25), Acc_Type VARCHAR(10),
```

```
  Amount DECIMAL(10,2)
) ;
```

Listing 3.10: Query:Creation of bank table

Inserting values into the *bank* table

```
INSERT INTO bank VALUES
('150001', 'Peter', 'Alappuzha', 'Current', '65000');

INSERT INTO bank VALUES
('150002', 'Hamid', 'Trivandrum', 'Savings', '12000');

INSERT INTO bank VALUES
('150003', 'Anoop', 'Malappuram', 'Savings', '4000');

INSERT INTO bank VALUES
('150004', 'Jyothi', 'Trissur', 'Current', '75000');

INSERT INTO bank VALUES
('150005', 'Azim', 'Ernakulam', 'Current', '58600');

INSERT INTO bank VALUES
('150006', 'Ayra', 'Kozhikkod', 'Savings', '7500');

INSERT INTO bank VALUES
('150007', 'Amosh', 'Trissur', 'Savings', '3500');

INSERT INTO bank VALUES
```

```
('150008', 'Sidharth', 'Kodungallore', 'Savings', '69200');
```

Listing 3.11: Query: Multiple queries to insert various records into the bank table.

```
Answer 1: SELECT * FROM bank WHERE Branch_Name="Kodungallore"
    ;
```

Acc_No	Acc_Name	Branch_Name	Acc_Type	Amount
150008	Sidharth	Kodungallore	Savings	69200.00

Figure 3.12: Account Details of Kodungallore branch

```
Answer 2: UPDATE bank SET Branch_Name ="Thiruvananthapuram"
    WHERE Branch_Name="Trivandrum";
Answer 3: SELECT * FROM bank WHERE Branch_Name IN
    ('Thiruvananthapuram', 'Kozhikkod', 'Ernakulam');
```

Acc_No	Acc_Name	Branch_Name	Acc_Type	Amount
150002	Hamid	Thiruvananthapuram	Savings	12000.00
150005	Azim	Ernakulam	Current	58600.00
150006	Ayra	Kozhikkod	Savings	7500.00

Figure 3.13: Account Details of Selected branches using IN clause

```
Answer 4: SELECT * FROM bank WHERE Branch_Name='Trissur'
    HAVING Amount < 5000;
```

Acc_No	Acc_Name	Branch_Name	Acc_Type	Amount
150007	Amosh	Trissur	Savings	3500.00

Figure 3.14: Account details of selected branches using HAVING clause

3.5 Create a *book* table and perform fundamental database operations

PROBLEM

Create a table book with the following fields and insert at least 5 records into the table.

Book_Id Integer Primary key, Book_Name Varchar (20),

Author_Name Varchar (25), Pub_Name Varchar (25), Price Decimal (10,2)

1. Display the details of book with price 100 or more.
2. Display the names of all books published by SCERT.
3. Increase the price of books by 10% which are published by SCERT.
4. List the details of books with the title containing the word "Programming" at the end.

SOLUTION

Creation of book table

```
CREATE TABLE book (
  Book_ID INT NOT NULL PRIMARY KEY,
  Book_Name VARCHAR(20),
  Author_Name VARCHAR(25),
   Pub_Name VARCHAR(25),
   Price DECIMAL(10,2));
```

Listing 3.12: Create Query

Inserting values to the table

```
INSERT INTO book VALUES ('1200', 'The Kite Runner', 'Khaled
   Husseini', 'River Head', '450');

INSERT INTO book VALUES ('1201', 'Wings of Fire', 'Dr. APJ
   Abdulkalam', 'Universities Press', '250');

INSERT INTO book VALUES ('1202', 'The White Tiger', 'Aravind
   Adiga', 'Atlantic', '300');

INSERT INTO book VALUES ('1203', 'Manju', 'M T Vasudevan Nair
   ', 'DC Books', '425');
INSERT INTO book VALUES ('1204', 'Arachar', 'K R Meera', 'DC
   Books', '225');

INSERT INTO book VALUES ('1205', 'When Breath Becomes Air', '
   paul Kalanidhi ', 'Random House', '525');

INSERT INTO book VALUES ('1206', 'Forty Rules of Love', 'Elif
    Shefak', 'Penguine', '700');

INSERT INTO book VALUES ('1207', 'Art of Computer Programming
   ', 'Peter ', 'SCERT', '125');

INSERT INTO book VALUES ('1208', 'Madhuram Malayalam', 'Madhu
    Vaduthala', 'SCERT', '75');

INSERT INTO book VALUES ('1209', 'Python Programming', '
   Michael Dawson', 'Pearson', '3300');
```

Listing 3.13: Query: Inserting various records into the *book* table

```
Answer 1: SELECT * FROM book WHERE Price >= 100;
```

Listing 3.14: Insert Query

Book_ID	Book_Name	Author_Name	Pub_Name	Price
1200	The Kite Runner	Khaled Husseini	River Head	450.00
1201	Wings of Fire	Dr. APJ Abdulkalam	Universities Press	250.00
1202	The White Tiger	Aravind Adiga	Atlantic	300.00
1203	Manju	M T Vasudevan Nair	DC Books	425.00
1204	Arachar	K R Meera	DC Books	225.00
1205	When Breath Becomes Air	paul Kalanidhi	Random House	525.00
1206	Forty Rules of Love	Elif Shefak	Penguine	700.00
1207	Art of Computer Programming	Peter	SCERT	125.00
1209	Python Programming	Michael Dawson	Pearson	3300.00

Figure 3.15: Books that costs 100 or more

```
Answer 2: SELECT * FROM book WHERE Pub_Name = 'SCERT';
```

Book_ID	Book_Name	Author_Name	Pub_Name	Price
1207	Art of Computer Programming	Peter	SCERT	125.00
1208	Madhuram Malayalam	Madhu Vaduthala	SCERT	75.00

Figure 3.16: Books that costs 100 or more

```
Answer 3: UPDATE book SET Price=Price+Price*0.1 WHERE
   Pub_Name = 'SCERT';

Answer 4: SELECT * FROM BOOK WHERE Book_Name LIKE '%
   Programming';
```

Book_ID	Book_Name	Author_Name	Pub_Name	Price
1207	Art of Computer Programming	Peter	SCERT	137.50
1209	Python Programming	Michael Dawson	Pearson	3300.00

Figure 3.17: Books that costs 100 or more

4 Environmental Configuration

4.1 C++ Programming with Geany IDE on Ubuntu Linux

C++ is a compiled language, utilizing a compiler to translate source code into object code. This object code is then transformed into an executable for end-users. To develop C++ code, an Integrated Development Environment (IDE) is necessary. In this guide, we will use Geany as our editor of choice. Several compilers are available for C++, including GCC, Borland C++, Turbo C++, Microsoft C++ (Visual C++), and Unix AT T C++. Of these, GCC is a free software available on Linux and Windows platforms. Geany is a versatile cross-platform IDE compatible with GCC for writing, compiling, and executing C++ programs.

Opening Geany IDE on Ubuntu Linux:

1. Access the Geany IDE from the Applications menu on Ubuntu Linux by following these steps: $Applications -> Programming -> Geany$
2. Create a new file by navigating to $File -> New$
3. In the new window, you can begin typing your C++ program. By

default, the file is named "untitled." To open a new file, go to the File menu, select the New option, or click the New button on the toolbar. Once you have a file open, enter your C++ program and save it with a suitable filename, including the *.cpp* extension.

4. To save your program, select the File menu and then the Save option or use the keyboard shortcut Ctrl+S.

Compiling and Running the Program:

Once the code is written, it needs to be compiled and executed:

Compile

To compile your program identify and correct errors if any are detected, follow these steps: $Build- > Compile$ a. If errors are present, they will be displayed in the compiler status window at the bottom. Otherwise, a message stating "Compilation finished successfully" will appear. b. After a successful compilation, choose the Build menu and select the Build option to link your program. $Build- > Build$

Running the Program:

Running a program involves executing the instructions defined in the code. To run your program:

Select the Build menu and choose the Execute option. Alternatively, you can execute the program by pressing the function key F5. $Build- > Execute$ By following these steps, you can create, compile, and run C++ programs

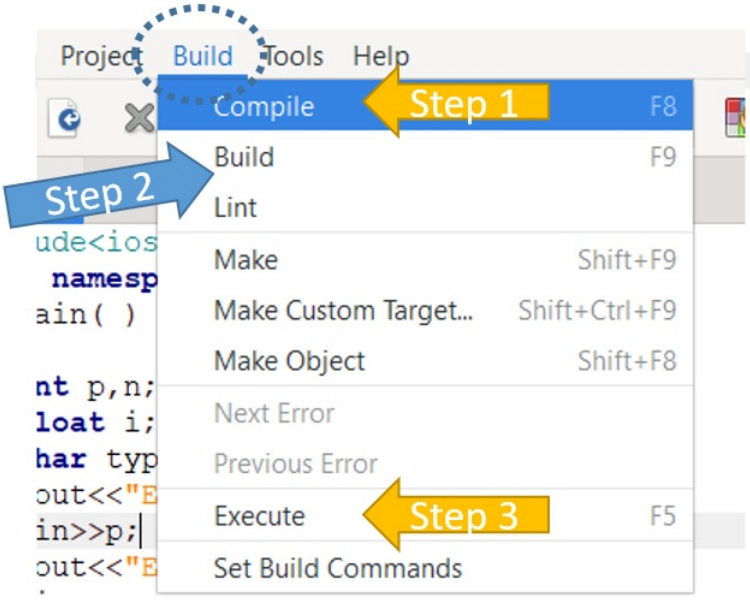


Figure 4.1: Steps to execute a C++ Program

using the Geany IDE on Ubuntu Linux.

4.2 HTML Tips - Creation and rendering

Using Geany Editor to create an HTML File:

To create an HTML file using the Geany Editor, follow these steps:

- Open Geany Editor.
- Start a new document and ensure the file extension is either *.html* or *.htm*.

HTML's Simplicity and Flexibility: HTML is known for its simplicity and flexibility. Unlike strongly typed languages, HTML is loosely typed, which means it doesn't require explicit data type declarations.

For example, you don't need to specify whether a value is a string or a number; HTML handles it based on context.

Tag Case Sensitivity: In HTML, tags are not case sensitive, which means you can use uppercase or lowercase letters for tags without affecting how the browser renders the content. However, when working with XHTML, all tags must be in lowercase. It's a good practice to use lowercase tags even when coding in HTML for consistency.

Proper Indentation: Maintaining proper indentation while writing HTML code is crucial. It helps you identify and locate unclosed or incorrectly typed tags more easily. This ensures your HTML code is well-structured and error-free.

HTML as a Markup Language: HTML is a markup language, which means it defines how content is displayed in a web browser. It does not support mathematical operations or scripting. Instead, it's parsed and rendered by the web browser to create the visual representation of a web page.

Viewing the HTML File: After creating the HTML file, you can locate it on your file system and open it in any web browser to view the web page you've created. Simply double-click on the HTML file, and your browser will render the content.

4.3 Getting Started with MySQL on Ubuntu Linux

MySQL is a relational database language designed for managing data, not to be confused with programming languages like C or C++. It provides various commands to create and manipulate tables, insert data, and perform data operations within tables.

To start working with MySQL on Ubuntu Linux, follow these steps:

1. Open the Terminal window by navigating through Applications ¿ Accessories ¿ Terminal.
2. Upon opening MySQL, it may prompt you for a password for verification. Use the same password you set during the installation process.
3. Once the password is verified, you will gain access to the MySQL prompt.

Before working with data, you need to create a database, which serves as a container for tables. To create a database in MySQL, use the following syntax:

```
CREATE DATABASE <database_name>;
```

Replace < *database_name* > with a meaningful and unique name for your database. To work with a specific database, you must explicitly open it. When you open a database, it becomes the active database within the MySQL server. MySQL provides the USE command for this purpose, with the following syntax:

```
USE <database_name>;
```

For example, to open the "school" database, use the command:

```
USE school;
```

After executing this command, you will receive the following response:

```
Database changed
```

This response indicates that you have successfully switched to the "school" database, and you are ready to work with the data within it.

To see the structure of created table use the *DESC* command. For example

```
DESC student;
```

Field	Type	Null	Key	Default	Extra
roll_number	int	NO	PRI	NULL	
stud_name	varchar(25)	YES		NULL	
batch	varchar(25)	YES		NULL	
mark1	int	YES		NULL	
mark2	int	YES		NULL	
mark3	int	YES		NULL	
total	int	YES		NULL	

Figure 4.2: Schema of the student table

Thank You

Code is like humor. When you have to explain it, it's bad

-Cory House