2019

Programming Fundamentals

(CS-101)

Lecture Notes
by
Dr. Minhaj Ahmad Khan

BS(CS) Semester-I, Session 2019-2023

Minhaj Ahmad Khan
Department of Computer Science
9/20/2019

Table of Contents

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Computer, Hardware and Software

It is an electronic device that takes input from input device, processes its and produces output on output device. It may also store data, code and output on storage devices.

Von Neumann Architecture

The basic computer system design using processing unit, control unit, memory, external storage and input-output (I/O) devices is based on Von Neumann Architecture. It uses the stored-program concept of a computer system in which instructions and data are stored in the same memory. The instructions are executed in a sequence in this architecture. The programs stored on a secondary storage device are brought to RAM and executed by CPU. The results may then be stored back in RAM. The high performance computers however allow for out-of-order execution of instructions.

Hardware

Physical components of a computer system are termed as hardware.

Software

A collection of programs is known as software. In general, a software may be categorized into:

1. Application Software
   e.g. Banking systems, campus management system etc
2. System Software
   e.g. Operating Systems, compilers

Solving Problems using Programs

A program is a set of instructions for the computer system to solve a particular problem. The programs are written in computer programming languages. The instructions of the programs are also known as statements in programming languages.

Programming Languages

The language in which computer programs are written is known as programming language.

Example: C++, Java, C#, Python, PHP, ASP, JSP, Visual Basic.NET etc.

In general, there are two types of languages:

1. High level Language
2. Low level Language

High level Languages

These are the languages which use English like form to write statements. These languages are easy to understand and program with.

Low level Languages

These languages are close to the hardware. It is difficult to program using low level languages. There are two types of low level languages:

1. Assembly Language
2. Machine Language

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Assembly Language

The Assembly Language programs use mnemonics as instructions, also known as operation code. Every CPU has different instruction set architecture (ISA), and therefore machine/assembly language of every CPU is different.

e.g. Add AX, BX

AX, BX are CPU registers, and Add is the instruction

Machine Language

The Machine Language consists of binary 0s and 1s. It is the language that is directly understood by the CPU. e.g.

00000010 0010 0011

Software Development Environments

Editor

Software that is used to create, save and modify programs.

IDE (Integrated Development Environment)

IDE is used to create, save, modify, compile and execute programs.

Example: Dev-C++, Visual Studio, Code Blocks, NetBeans etc.

Source Code

The code which is written by the programmer in a computer language, is called source code.

Object Code

The code that is generated by the compiler is termed as the object code.

Compiler

It is a translator that converts source code into object code. The object code is usually a machine language code which may be executed by CPU.

Source code ----------> Object Code (Usually machine code)

Examples: GNU C compiler, g++, gcc, Turbo C++ compiler tcc, Visual C compiler vc

Linker

The linker is used to link all the compiled modules of a program and generates an executable program.

Assembler

The assembler converts a program in assembly language to machine language code.

Compilation Procedure

a.cpp ---------> a.exe

a.cpp ---------> a.s ---------> a.obj ---------> a.exe

a.cpp ---------> a.s ---------> a.out ---------> a.out

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Linking

a.cpp ---------> a.s ---------> a.obj ---------> a.exe

b.cpp ---------> b.s ---------> b.obj

g++ -o a.exe a.obj b.obj

Debugger

The debugger is used to execute program on line by line basis to remove error.

Pre-Processor

A preprocessor is a part of compiler and it is used to process preprocessors directives.

Pre-Processor Directives

These are the instructions for the preprocessors to perform some functionality

- #include
  - To include contents of other files
- #define
  - To define literals for replacement in code
- #if

Syntax

The syntax of language is the format in which programs should be written. Every language has a syntax which should be followed. If the programmer does not follow the syntax, the compiler generates the syntax error and executable file is not generated.

For example, for assignment, the syntax is:

variable '=' expression

Using this format, we can write statements e.g. A = 5*3

Programming Language Basic Terms

Constants

Memory location where values cannot be changed during execution of program, are termed as constants.

Example: const i = 7;

Variables

Memory locations where values can be changed during execution of the program.

Example: int i =7;i=10;

Keywords

Keywords are the special words defined by the language. These are reserved for its own use and have special purpose in the language.

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Identifiers

Identifiers are programmer defined words for variables, constants etc. The identifier name must start with alphabet or underscore. It cannot contain special symbols or operators. It can contain combination of digits, underscores, alphabets, after first character.

Statement

It is an instruction of the program. In C++, it is terminated by a semicolon.

e.g. A = 10;

cout << "A=" << A << endl;

Indentation

The indentation refers to the spaces and margins given while writing program statements. This is required for better readability of code. Extra spaces are ignored by the C++ compiler.

Literals

Literals are the constant values used in the program.

e.g: 7 8.59 “abc”

String Literals

A collection of alpha numeric characters including special characters enclosed within double quotation marks are termed as string literals.

e.g. “abc 12”

Comments

Comments are the lines for programmers' understanding of code. These lines are ignored by the compiler. There are two types of comment in C++.

```
// Single line comment

/* This is a

        multiline comment  */
```

Compound Statement

Collection of statements enclosed in braces is collectively termed as a compound statement.

```
{
 cout <<”First” << endl;
  cout  <<”Second” << endl;
}
```

Data Types

A data type defines the values that a variable of that type can have.

Ex: int, float, double, char, short, long etc.

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Size of Data Types and their Range (g++)

The data types can be signed or unsigned. For unsigned data types, we have to use the "unsigned" keyword.

Un-signed Data Types

unsigned char = 1 byte ----------> 0 to 2^8 - 1

unsigned short = 2 bytes ------→ 0 to 2^{16} - 1

unsigned int = 4 bytes

unsigned long = 8 bytes

e.g. unsigned int B = 5;

Signed Data Types

1-bit is used for sign, and the remaining bits are used for value. Consequently, the range becomes:

char ----> 1 byte ----> (-2^7 to 2^7-1)

short----> 2 byte ---->(-2^{15} to 2^{15}-1)

int short----> 4 byte ----->(-2^{31} to 2^{31}-1)

long short----> 8 bytes ----->(-2^{63} to 2^{63}-1)

float --> single precision real numbers

double--> double-precision real numbers

e.g. double A = 3.4;

Char Data Type

The char data type is used for processing individual character. In C language, the character value is enclosed within the apostrophes.

char ch = '0';

cout<<ch<<endl;

C++ Operators and their Precedence

Group wise precedence

:: Scope resolution operator

. Member selection operator, ->Member selection operator (pointer), postfix ++ -- , [] ()

++ -- prefix operators, & Address-of, + - ! ~ Unary operators

* / % binary operator

+ - binary

<< left shift, >> right shift

< > <= >= comparison all inequalities

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

== !=

& bitwise AND

^ bitwise exclusive OR

| bitwise OR

&& logical AND

| logical OR

?: conditional ternary operator

= assignment, *= /= -= <<= >>= &= != ^=

Algorithm and its Implementation

A step by step procedure to solve a particular problem is termed as Algorithm. It uses English like form to describe the steps, which is called pseudo-code.

Write an algorithm to input two numbers and display their sum:

1- Begin
2- Input A
3- Input B
4- Compute C=A+B
5- Print C
6- End

Program Implementation in C++

```
#include<iostream> // header file for input-output stream (cin,cout)

#include<conio.h> // header file for console input-output (getch)

using namespace std; // namespace for cin, cout

int main() // main function required for each C++ program

{

    int a,b,c; // Type declaration of variables

    cout<<"Give number a"<<endl; // output statement using stream-insertion operator <<

    cin>>a; // input statement using stream-extraction operator >>

    cout<<"Give Number B"<<endl;

    cin>>b;

    c=a+b; // Assignment statement for computation

    cout<<"Sum is"<<c << endl;

    getch();

    return 0; // value to be returned by the main function
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

}

Write an algorithm to input two numbers a and b, and display the result of the following expression: $C = A^2 + 2B$

1- BEGIN
2- INPUT A
3- INPUT B
4- COMPUTE C = (A*A) +(2*B)
5- PRINT C
6- END

Flowchart

A flowchart is used to depict the steps of an algorithm. It makes use of the following special symbols.

- Rounded Rectangle 

It is used to depict the beginning and ending of algorithm.

-  Rectangle

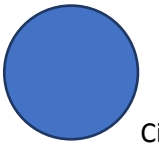
It is used to represent the computations performed by the algorithm.

-  Parallelogram

It is used to represent input and output steps of the algorithm

-  Diamond

The diamond symbol is used to represent conditional statements.

-  Circle

It is used to combine multiple flows of control.

-  Edges or Arrows

It represents the flow of control in the algorithm.

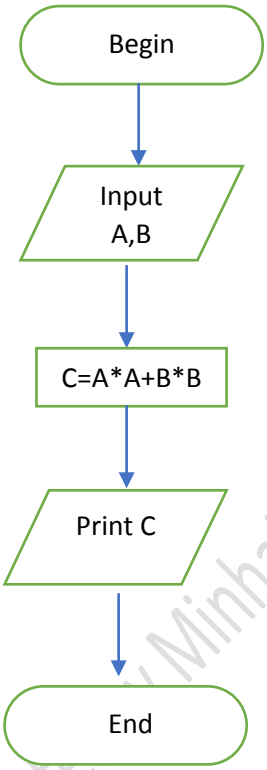
Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Write algorithm, flow chart and program in C++ to input two numbers and display the sum of their squares.

Algorithm

1- BEGIN
2- INPUT A
3- INPUT B
4- COMPUTE C = A*A + B*B
5- PRINT C
6- END

Flow Chart



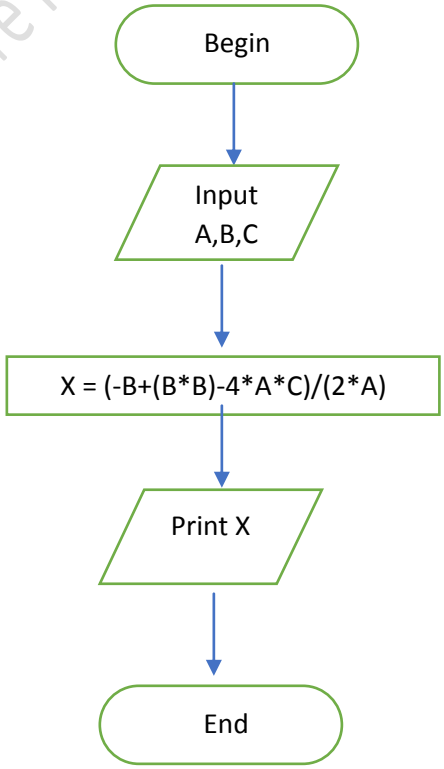
Write algorithm, flow chart and program in C++ to input three numbers a,b,c and display the result of following expression:

$(-b+b^2-4ac)/2a$

Algorithm

1- BEGIN
2- INPUT A
3- INPUT B
4- INPUT C
5- COMPUTE X = (-B+(B*B)-4*A*C)/(2*A)
6- PRINT X
7- END

Flow Chart



Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Conditional Statements

The conditional statements make use of condition for execution of statements.

If-Statement

Syntax

```
if (condition)
        Statement;
if (condition)
        Statement;
else
        Statement;
```

Nested If-Statement

The if statement embedded within another if statement is termed as a nested if-statement.

```
If ( a>0)
        if (b>0)
                cout<<"a and b are Positive"<<endl;
        else
                cout<<"a is Positive, b is Negative"<<endl;
```

If-Else-If

```
if (a>=5 && a<=10)
        cout<<" 5 to 10"<<endl;
else if (a>=2 && a<=4)
        cout<<"2 to 4" <<endl;
else if (a >=0 && a<=1)
        cout<<" 0 to1 "<<endl;
else
        cout<<"Negative"<<endl;
```

Write a program to input a number and display whether it is positive, negative or zero.

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
        int a;
        cout<<"Give a "<<endl;
```

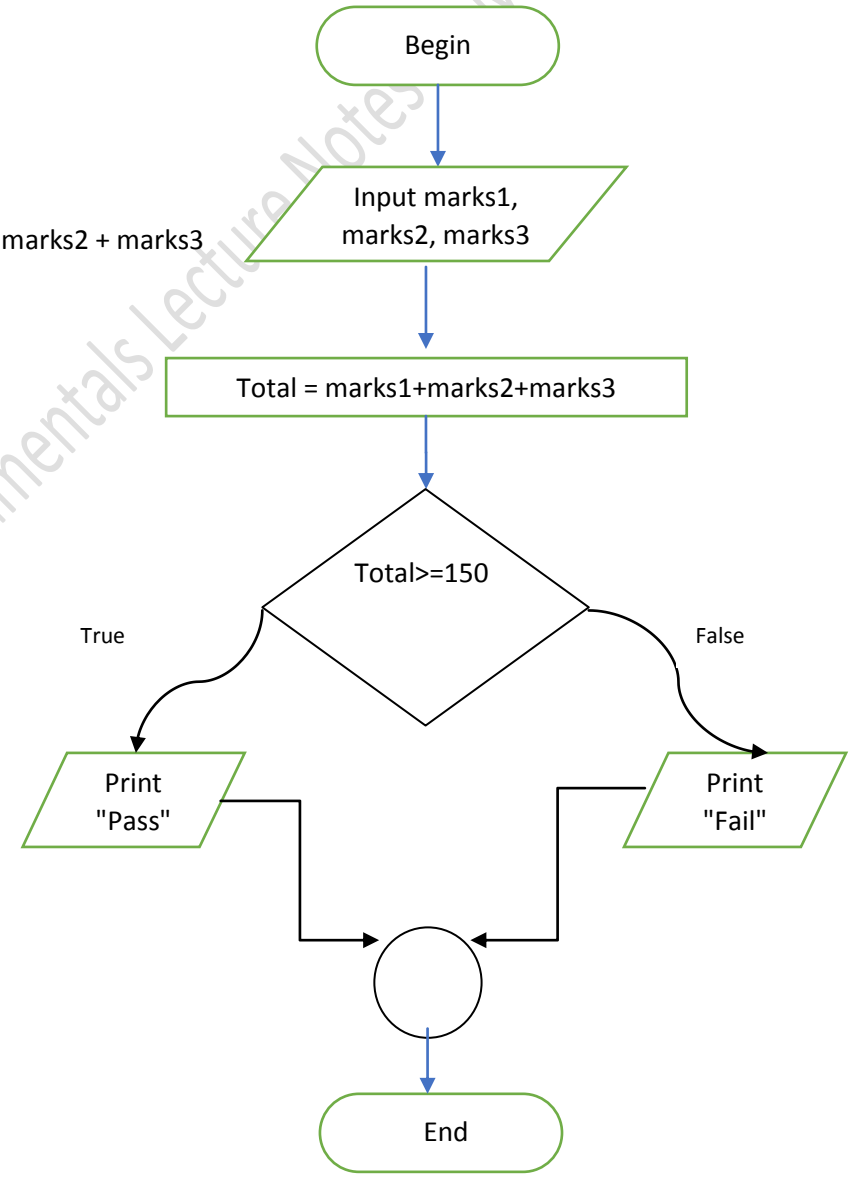
Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
        cin>>a;
        if (a<0)
            cout<<"Negative"<<endl;
        else if (a>0)
            cout<<"Positive"<<endl;
        else
            cout<<"Zero"<<endl;
        getch();
        return 0;
}
```

Write algorithm, flowchart and program in C++ to input marks of 3 subjects of a student and display whether he has passed or failed the exam. (Assuming total>=150 to pass the exam)

Algorithm

1- BEGIN
2- INPUT marks1
3- INPUT marks2
4- INPUT marks3
5- COMPUTE total = marks1 + marks2 + marks3
6- IF total >= 150 Then
    PRINT "pass"
7- ELSE
    PRINT "Fail"
8- END IF
9- END



Programming Fundamentals Lecture Notes Minhaj Ahmad Khan

Assignments

Write algorithm, flowchart and program in C++ to input two numbers a,b and display which one is greater or they are equal.

Write algorithm, flowchart and program in C++ to input marks of three subjects of a student and display his grade according to given criteria.

Percentage	Grade
>=80	A
>=70 and <80	B
>=60 and <70	C
>=50 and <60	D
<50	F

Switch Statement

```
switch( variable )
{
case val1:
        statement;
        break;
case val2:
        statement;
        break;
default:
        statement;
}
```

Break Statement

The break statement is used to exit from the switch statement. It is also used to transfer control out of the blocks of the loop.

The general format is: break [label];

Continue Statement

The continue statement is used to start next iteration of a loop. See example in loops.

Conditional Operator

The operator ?: is a ternary operator since it takes three operands.

```
c = (a>b) ? 10:20;
cout<<" c= "<<c<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Assignment:

Write algorithm, flow chart and program in C++ to input three numbers a,b,c and display the highest value.

Loops

Loops are used to execute statement iteratively. There are three types of loops in C++.

1- For Loop
2- While Loop
3- Do-While Loop

For Loop

for (initialization; condition; increment/decrement)

Statement;

Ex:

```
int i;
for ( i = 1; i<10; i++)
{
    cout<<i<<endl;
}
```

Write a program to display odd numbers from 1 to 99

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int i;
    for(i=1;i<100;i=i+2)
        cout<<"i="<<i<<endl;
    getch();
    return 0;
}
```

Write a program to print even numbers 100 to 2 in descending order:

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

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
using namespace std;
int main()
{
    int i;
    for(i=100;i>=2;i=i-2)
        cout<<"i="<<i<<endl;
    getch();
    return 0;
}
```

Write a program to display sum of the following series 1+2+3--------+100

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int i,sum=0;
    for(i=1;i<=100;i=i+1)
        sum=sum+i;
    cout<<"Sum is"<<sum<<endl;
    getch();
    return 0;
}
```

Write a program to display the sum of the following series 2+4+6-------+100

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int i,sum=0;
    for(i=2;i<=100;i=i+2)
        sum=sum+i;
    cout<<"Sum is"<<sum<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    getch();
    return 0;
}
```

Write a program to display the product of following series 2*4*6-------*100

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int i, pro=1;
    for(i=2;i<=100;i=i+2)
        pro=pro*i;
    cout<<"Product="<<pro << endl;
    getch();
    return 0;
}
```

Write a program to display sum of following series 1-2+3-4+---------- -100

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int i,sum=0,sign=1;
    for(i=1;i<=100;i=i+1)
    {
        sum=sum+i*sign;
        sign=sign*(-1);
    }
    cout<<"Sum is"<<sum;
    getch();
    return 0;
}
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Write the program to input an number and display the result of the following series 1+2--------+n

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int I,sum=0,n;
    cout<<"Give N"<<endl;
    cin>>n;
    for(i=1;i<=n;i=i+1)
    {
        sum=sum+i;
    }
    cout<<"Sum is"<<sum << endl;
    getch();
    return 0;
}
```

Write a program to display the sum of following series:

(1+2)+(2+3)+----------+(99+100)

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int i,sum=0;
    for(i=1;i<=99;i=i++)
        sum=sum+(i+(i+1));
    cout<<"Sum is"<<sum << endl;
    getch();
    return 0;
}
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Assignment

Write a program to input a number n and display the result of the following series:

(1+2)-(2+3)+---------- ±(n+(n+1))

(1-2)+(2+3)+----------+(n±(n+1))

While Loop

```
while(condition)
    Statement;
```

Example:

```
i=1;
while(i<10)
{
        cout<<i<<endl;
        i++;
}
```

Write a program using while loop to print even numbers from 2 to 100.

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int I = 2;
    while (I < = 100)
    {
        cout<<"I = " <<i <<endl;
        i = i +2;
    }
    getch();
    return 0;
}
```

Do-While Loop

```
do {
        Statements;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
} while( condition );

i = 1;

do {

    cout<<"i="<<i<<endl;

    i = i+1;

} while (i<=10);
```

Write code to print values from to 2 to 100 (even) using do while loop.

```
#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

    int I;

    i = 2;

    do {

        cout<<ï="<<i<<endl;

        i= i+2;

    } while(i<=100);

    getch();

    return 0;

}
```

Assignments

Write a program to input a number and display the sum of the series **1-2+3- ----- ± n** using for-loop, while-loop and do-while-loop.

Nested Loops

A loop embedded within another loop is termed as a nested loop.

Example

```
for(i=0;i<2;i++)

{

    for(j=0;j<3;j++)

    {
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
            for(k=0;k<1;k++)
            {
                  cout<< ”i=”<<i<<”j=”<<j<<”K=”<<k<<endl;
            }
      }
}
```

Using Break and Continue in Loops

```
for(i=0;i<100;i++) // Loop printing numbers from 0 to 9
{
      if (i>= 10) break;
       cout<<ï=”<<i<<endl;
}
for(i=0;i<100;i++) // Loop to print odd numbers
{
      if ((i%2) == 0) continue;
       cout<<ï=”<<i<<endl;
}
```

Arrays

Arrays are homogeneous data structures whose elements are stored at consecutive memory locations.

Declaration Example

```
int a[10]; // 10 integers, 40 bytes of memory
```

This array has 1-dimension.

Programs using 1-dim Arrays

Write a program to input and output 10 elements of an array of integers.

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    int a[10];
    cout<<"Give data"<<endl;
    int I;
    for (i=0; i<10;i++)
    {
        cin>>a[i];
    }
    cout<<"You entered"<<endl;
    for (i=0; i<10;i++)
    {
        cout<<a[i]<<endl;
    }
    getch();
    return 0;
}
```

Write a program to input 10 elements of array and display their sum.

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int a[10], sum =0;
    cout<<"Give data"<<endl;
    int i;
    for (i=0; i<10;i++)
    {
        cin>>a[i];
    }
    cout<<"You entered"<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    for (i=0; i<10;i++)
    {
        sum = sum + a[i];
    }
    cout<<"sum is "<<sum<<endl;
    getch();
    return 0;
}
```

Write a program to input 10 elements of array and display sum of even numbers in the array.

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int a[10], sum = 0;
    cout<<"Give data"<<endl;
    int i;
    for (i=0; i<10;i++)
    {
        cin>>a[i];
    }
    cout<<"You entered"<<endl;
    for (i=0; i<10;i++)
    {
        If(a[i]%2 == 0)
        {
            sum = sum + a[i];
        }
    }
    cout<<"sum is "<<sum<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    getch();
    return 0;
}
```

Assignment

1. Write code to input an array of 10 elements and display the sum of even and odd numbers separately.
2. Write code to input two arrays of 10 elements each, place the sum at corresponding elements in 3rd array and display the result.

Write a program to input array of 10 elements and display the highest value together with its location.

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int a[10], highest,loc;
    cout<<"Give data"<<endl;
    int i;
    for (i=0; i<10;i++)
    {
        cin>>a[i];
    }
    highest=a[0]; loc=0;
    for (i=0; i<10;i++)
    {
        If(a[i]>highest)
        {
            highest=a[i];
            loc=i;
        }
    }
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    cout<<"Highest is "<<highest<<endl;
    cout<<"Loc is"<<loc<<endl;
    getch();
    return 0;
}
```

Assignment

Write a program to input an array of 10 elements and display the lowest value together with its location.

Multi-Dimensional Arrays

It is possible to declare multi-dimensional arrays. Each dimension has a size.

Declaration: int a[3][2]

A matrix is represented as a 2-dim array.

Write a program to input a matrix of order 3*4 and display the input data

```
#include<iostream>
using namespace std;
#include<conio.h>
int main ()
{
    int i,j,a[3][4];
    cout<<"enter the values"<<endl;
    for(i=0;i<3;i++)
    {
        for(j=0;j<4;j++)
        {
            cin>>  a[i][j];
        }
    }
    cout<<"You Enterd"<<endl;
    for(i=0;i<3;i++)
    {
        for(j=0;j<4;j++)
        {
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
            cout<<a[i][j] << "\t";
        }
        cout << endl;
    }
    getch ();
    return 0;
}
```

Write the program to input an array of order 2*3*4 and display the input data.

```
#include<iostream>
using namespace std;
#include<conio.h>
int main ()
{
    int i,j,a[2][3][4];
    cout<<"enter the values"<<endl;
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
        {
            for(k=0;k<4;k++)
            {
                cin>>  a[i][j][k];
            }
        }
    }
    cout<<"You Enterd"<<endl;
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
        {
            for(k=0;k<4;k++)
            {
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
                cout<<a[i][j];
            }
        }
    }
    getch ();
    return 0;
}
```

Write the program to input two matrices of order 3*4 and display their sum.

```
#include<iostream>
using namespace std;
#include<conio.h>
int main ()
{
    int i,j,a[3][4],b[3][4],c[3][4];
    cout<<"give data of 1st matrix"<<endl;
    for(i=0;i<3;i++)
    {
        for(j=0;j<4;j++)
        {
            cin>>a[i][j];
        }
    }
    cout<<"give data of 2nd matrix"<<endl;
    for(i=0;i<3;i++)
    {
        for(j=0;j<4;j++)
        {
            cin>>b[i][j];
        }
    }
    cout<<"3rd matrix result "<<endl;
    for(i=0;i<3;i++)
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    {
        for(j=0;j<4;j++)
        {
            c[i][j]=a[i][j]+b[i][j];
            cout<<c[i][j]<<" ";
        }
        cout<<endl;
    }
    getch();
    return 0;
}
```

Write a program to input a matrix of order 3*4 and display the sum of each row and column separately.

```
#include <iostream>
using namespace std;
#include <conio.h>
int main()
{
    int arr[3][4];int tr=0,tc=0,c=4,r=3;
    /* cout<<"Enter the number of rows and columns."<<endl;
    cin>>r>>c; */
    cout<<"Enter the elements of array."<<endl;
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            cin>>arr[i][j];
        }
    }
    for (int i = 0; i < r; i++)
    {   tr =0;
        for (int j = 0; j < c; j++)
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
        {
            tr=tr+arr[i][j];
        }
        cout<<"SumR="<<tr<<endl;
    }
    for (int j = 0; j < c; j++)
    {   tc=0;
        for (int i = 0; i < r; i++)
        {
            tc=tc+arr[i][j];
        }
        cout<<"SumC="<<tc<<endl;
    }
    getch();
    return 0;
}
```

Write a program to input a matrix of order 4*4 and display the highest and lowest number in each row separately.

```
#include<iostream>
using namespace std;
#include<conio.h>
int main ()
{
    int i,j,a[4][4],highest,lowest;
    cout<<"give data of an array"<<endl;
    for(i=0;i<4;i++)
    {
        for(j=0;j<4;j++)
        {
            cin>>a[i][j];
        }
    }
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    for(i=0;i<4;i++)
    {
        highest=a[i][0];
        lowest=a[i][0];
         for(j=0;j<4;j++)
        {
            if(a[i][j] > highest)
                highest=a[i][j];
            if(a[i][j] < lowest)
                lowest=a[i][j];
        }
        cout<<"highest is"<<highest<<endl;
        cout<<"lowest is"<<lowest<<endl;
    }
    getch ();
    return 0;
}
```

Assignment

Write a program to input matrix 4*4 and perform the followings:

1. Display the smallest number in every row separately
2. Display the highest and lowest values in every column separately
3. Display the sum of diagonals separately
4. Display the highest and lowest values of diagonals separately

Write a program to input two matrices of order 3*4 and 4*3 respectively. Display the result of the product of these matrices.

Structures in C++

The structures in C++ represent heterogeneous data structures. A structure can contain more than one element, each with possibly different data type.

Write a program to input two student(rollno, marks) structures and display the input data.

```
#include<iostream>
using namespace std;
#include<conio.h>
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
struct Student
{
    int rollno;
    int marks;
};
int main()
{
    Student s1, s2;
    cout<<"Give data of 1st student"<<endl;
    cin>>s1.rollno; // dot is membership operator
    cin>>s1.marks;
    cout<<"Give data of 2nd student<<endl;
    cin>>s2.rollno;
    cin>>s2.marks;
    cout<<"Data of student 1"<<endl;
    cout<<"Rollno ="<s1.rollno<<endl;
    cout<<"marks="<<s1.marks<<endl;
    cout<<"Data of student 2"<<endl;
    cout<<"Rollno"<<s2.rollno<<endl;
    cout<<"Marks"<<s2.marks<<endl;
    getch();
    return 0;
}
```

Write a program to input an array of 10 product (Pid, Pname, Qty) structures and display the input data.

```
#include<iostream>
using namespace std;
#include<conio.h>
#include<string.h>
struct Product
{
    int PId;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    char Pname[30];
    int Qty;
};
int main ()
{

    Product p[10];
    int i;
    cout<<"give the data"<<endl;
    for(i=0;i<10;i++)
    {
        cin>>p[i].PId;
        cin>>p[i].Pname;
        cin>>p[i].Qty;
    }
    cout<<"you entered"<<endl;
    for(i=0;i<10;i++)
    {
        cout<<"PId ="<<p[i].PId<<endl;
        cout<<"Pname ="<<p[i].Pname<<endl;
        cout<<"Qty ="<<p[i].Qty<<endl;
    }
    getch ();
    return 0;
}
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Assignment

- Write a program to input an array of 10 student (rollno, name, marks) structures and display the input data.
- Write a program to input an array of 10 book (bid, btitle, price) structures and display the input data.

Write a program to input an array of 10 product (pid, pname, qty) and display data of the product having the highest quantity.

```
#include<iostream>
using namespace std;
#include<conio.h>
#include<string.h>
struct product
{
    int PId;
    char Pname[30];
    int Qty;
};
int main ()
{
    product p[10];
    int i;
    cout<<"give the data"<<endl;
    for(i=0;i<10;i++)
    {
        cin>>p[i].PId;
        cin>>p[i].Pname;
        cin>>p[i].Qty;
    }
    cout<<"you entered"<<endl;
    int high = p[0].qty, loc = 0;
    for(i=0;i<10;i++)
    {
        If(p[i].qty > high)
        {
            high = p[i].qty;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
            loc = i;
        }
    }
    cout<<"PId ="<<p[loc].PId<<endl;
    cout<<"Pname ="<<p[loc].Pname<<endl;
    cout<<"Qty ="<<p[loc].Qty<<endl;
    getch ();
    return 0;
}
```

Pointers and Memory Allocation

Pointers are memory locations that contain addresses of other memory locations.

```
int a; // static memory allocation
int *ptr;
ptr = &a;
a = 7;
*ptr = 10;
cout<<"a="<<a<<endl;
```

Indirection and Multiple Indirection

The values being pointed to by pointers may be accessed using * (value) operator. In multiple indirection, the value may further be a pointer to another address.

Pointer Constants and Pointer Variables

Pointer constant is the constant whose value can not be modified, whereas the value of pointer variable can be modified. The pointer constants are different from pointers to constants.

Arrays are pointer constants which use static memory allocation.

```
int a[30]; // static memory allocation of 120 bytes
int *a;
```

The pointers are used to perform dynamic memory allocation and de allocation. In **C++,** the "new" keyword is used for dynamic memory allocation and "delete" keyword is used for dynamic memory de allocation.

```
int *a;
a = new int [10]; // dynamic memory allocation
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
delete []a;
```

Pointers to Structures

Write a program to create and input data of 10 product (pname, pid, qty) structures using pointers and display the input data.

```
#include<iostream>
using namespace std;
#include<conio.h>
#include<string.h>
struct Product
{
    int PId;
    char Pname[30];
    int Qty;
};
int main ()
{
    Product * p = new Product [10];
    int i;
    cout<<"give the data"<<endl;
    for(i=0;i<10;i++)
    {
        cin>>(p+i)->PId;
        cin>>(p+i)->Pname;
        cin>>(p+i)->Qty;
    }
    cout<<"you entered"<<endl;
    for(i=0;i<10;i++)
    {
        cout<<"PId ="<<(p+i)->PId<<endl;
        cout<<"Pname ="<<(p+i)->Pname<<endl;
        cout<<"Qty ="<<(p+i)->Qty<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    }
    getch ();
    return 0;
}
```

Assignment

- Write the program to input data of 10 books(Bid,Title,Price) structures using pointers and display the input data

Write a program to input data of 10 student (Rollno, name, marks) structures using pointers and display the record of student having the highest Marks.

```
#include<iostream>
using namespace std;
#include<conio.h>
#include<string.h>
struct student
{
    int rollno;
    char name[30];
    int marks;
};
int main ()
{
    student*s;int i,j,highest;
    s=new student [10];
    cout<<"give data of student "<<endl;
    for(i=0;i<10;i++)
    {
        cin>>(s+i)->rollno;
        cin>>(s+i)->name;
        cin>>(s+i)->marks;
    }
    cout<<"you entered"<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
        highest=(s+0)->marks;
        int  loc=0;
        for(i=0;i<10;i++)
        {
            if(highest<(s+i)->marks)
            {
                highest=(s+i)->marks;
                loc=i;
            }
        }

        cout<<"data of student having highest marks"<<highest<<endl;
        cout<<"roll.no ="<<(s+loc)->rollno<<endl;
        cout<<"name ="<<(s+loc)->name<<endl;
        cout<<"marks ="<<(s+loc)->marks<<endl;
        delete[]s;
        getch ();
        return 0;
    }
```

Assignment

- Write a program to input data of 10 product (PID,Pname,QTY) structures using pointers and display data of Product having the lowest quantity.

Arrays as Members of Structures

Write the program to input 10 student(rollno, name, marks[6]) structure using pointer and display the input data.

```
#include<iostream>
using namespace std;
#include<conio.h>
#include<string.h>
struct student
{
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
        int rollno;
        char name[30];
        int marks[6];
};
int main ()
{
        student*s;int i,j;
        s=new student [10];
        cout<<"give data of student "<<endl;
        for(i=0;i<10;i++)
        {
            cin>>(s+i)->rollno;
            cin>>(s+i)->name;
            for(j=0;j<6;j++)
            {
                cin>>(s+i)->marks[j];
            }
        }
        cout<<"you entered"<<endl;
        for(i=0;i<10;i++)
        {
            for(j=0;j<6;j++)
            {
                cout<<"roll.no ="<<(s+i)->rollno<<endl;
                cout<<"name ="<<(s+i)->name<<endl;
                for (j=0;j<6;j++)
                {
                    cout<<"marks ="<<(s+i)->marks[j]<<endl;
                }
         }
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    delete[]s;
    getch ();
    return 0;
}
```

Assignment

- Write the program to input 10 student(rollno, name, marks[6]) structures using pointers and display data of the student having the highest total marks.

Strings in C++

It is collection of characters. In C++, it may be declared as an array of characters.

```
char str1[100];
```

There is a data type string which also exists in latest versions of C++.

```
string name;
```

Null character

Every string in C++ contains the null characters as the last character. Also known as '\0'.

Using C-Style String Functions

```
#include<iostream>
using namespace std;
#include<conio.h>
#include<string.h>
int main ()
{
    char str1[30];
    strcpy(str1,"firstabcd");
    cout<<str1[0]<<endl;
    cout<<strlen(str1)<<endl;
    cout<<strcmp("abc","Abc")<<endl;
    cout<<str1[strlen(str1)-2]<<endl;
    getch ();
    return 0;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

}

Functions and Modular Programming

In modular programming, a program may be divided into several modules. In C++, we can divide program into several files with each file containing functions.

The functions are named blocks of code . A function is used to perform specific functionality. In modular programming, it is a basic unit.

Structured Programming -- using high-level language constructs

Modular Programming -- A program is divided into several modules.

In C++, a program that does not contain main function can't be executable but it can be compiled.

Parameter Passing

The parameters or arguments may be passed to a function. When a function is called, the control transfers to the function body, and upon return the control transfers back to the calling point, to execute next statement after the function call. There are two types of parameters: actual parameters, formal parameters.

Actual Parameters

The parameters at the calling point are actual parameters.

Formal Parameters

The parameters at the function declaration body are formal parameters.

The parameters may be passed by value or by reference.

By-Value Parameter Passing

In by-value parameter passing, the value of the actual parameter is copied to the formal parameter. So any change to the formal parameter is not reflected back at the calling point.

By-Reference Parameter Passing

In by-reference parameter passing, the reference/address of the actual parameter is passed to the formal parameter. So any change to the formal parameter is reflected back at calling point.

Program

```
#include<iostream>

using namespace std;

#include<conio.h>

#include<string.h>

void function1(int &x1,int x2)

{
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    cout<<"yes"<<endl;
    x1=10;x2=7;
    cout<<"x1 ="<<x1<<endl;
    cout<<"x2 ="<<x2<<endl;
};
int main ()
{
    cout<<"main started"<<endl;
    int y=6;
    int z=4;
    function1(y,z);
    cout<<"y ="<<y<<endl;
    cout<<"z ="<<z<<endl;
    getch ();
    return 0;
}
```

Role of Stack in Parameter Passing

The function calls and parameter passing is implemented by compilers by using stack, a last-in-first-out data structure. The calling point (address) of a function and parameters are pushed on the stack and later retrieved or popped off the stack upon return.

Scope and Lifetime of Variables

The scope of a variable represents the places of code where a variable can be used. Lifetime is time when a variable continues to exist.

Auto Variables

The scope of variable starts from the point where it is declared and it covers all the sub-blocks of the code. An auto variable continues to exist till completion of the block in which it is declared.

```
{   int x;
    {
    .....
    }
}
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Static Variables

A static variable has the scope of function in which it is declared while it has the lifetime of the entire program.

The static Variables are initialized once but they continue to exist during multiple call of function so they retain their previous values. The static variables are declared using the static keyword.

static int i = 0;

Example

```
using namespace std;
#include<iostream>
#include<conio.h>
void function1();
void function2()
{
    static int i=0;
    cout<<"I="<<i<<endl;
    i++;
}
int main()
{
    cout<<"Now you are in the main function"<<endl;
    function1((();
    function1();
    function1();
    getch();
    return 0;
}
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Local Variables

Variables declared local to a block.

Non-Local Variables

Variables declared in enclosing block.

Global Variables

Variable declared outside the block and accessible everywhere in program. The lifetime of global variable is the complete program.

External Variables

The external variables are used to access global variables declared in other files. These are delcared using the extern keyword.

```
extern int i; // tells the compiler to
```

Recursion

Recursion is the feature of programming languages through which a function may call itself directly or indirectly. Such functions are termed as recursive functions. The recursive functions are easy to implement, but these function are slower than iterative implementations.

Program

```
#include<iostream>
using namespace std;
int fact(int n)
{
    if(n==0)
    {
        return 1;
    }
    else
    {
        return (n)*(fact(n-1));
    }
}
int main ()
{
    int n;
    cout<<"ENTER A NUMBER = ";
    cin>>n;
    fact(n);
    int k=fact(n);
```

Programming Fundamentals Lecture Notes by Minhaj Ahmed Khan

```
    cout<<k<<endl;
    getch();
    return 0;
}
```

Write a program using recursive function to print values from number n to 1.

```
#include<conio.h>
#include<iostream>
void disp(int n);
int main()
{
    int n;
    cout<<"Give n"<<endl;
    cin>>n;
    disp(n);
    getch();
    return 0;
}
void disp(int n)
{
    if(n==0)
        return;
    else
    {
        cout<<n<<endl;
        n--;
        disp(n);
    }
}
```

Write a program to display numbers from 2 to n even numbers using recursion.

```
#include<iostream>
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
Using namespace std;
#include<conio.h>
void disp(int n,int m);
int main()
{
    int n;
    cout<<"Give n"<<endl;
    cin>>n;
    disp(2,n);
    getch();
    return 0;
}
void disp(int m,int n)
{
    if(m==n)
    {
        cout<<m<<endl;
        return;
    }
    else
    {
        cout<<m<<endl;
        m=m+2;
        disp(m,n);
    }
}
```

Assignment

- Write a program to print odd numbers from n to 1 using recursion(assuming N is odd)

Prototypes of Functions

A prototype describes the return type, function name and arguments to the compiler.

```
void function1(int, int);
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

The compiler is able to compile code without any warning/error if it finds prototype without function body. Any functions in the prototype, if not found during compilation, are searched during linking of the program modules.

File Handling in C++

File

A file contains data on some storage device. There are two types of files in C++: text files, binary files.

Text Files

Text files contain data in the form of alphabetic characters (textual form) that are understandable to humans.

Binary Files

A binary file contains data in binary form (using characters representing values produced through combination of 0's and 1's) which are not easily understandable for humans.

Writing Data to a Text File

```
#include<iostream>

using namespace std;

#include<conio.h>

#include<fstream>

int main ()

{

    ofstream ofs("f1.txt");

    ofs<<"first line"<<endl;

    ofs<<"second line"<<endl;

    ofs<<"third line"<<endl;

    ofs.close();

    getch ();

    return 0;

}
```

Reading Data from a Text File

```
#include<iostream>

using namespace std;

#include<conio.h>

#include<fstream>

int main ()
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
{
    ifstream ifs("f1.txt");
    char str1[80];
    while(1)
    {
        ifs.getline(str1,80);
        if(ifs.eof())
            break;
        cout<<str1<<endl;
    }
    ifs.close();
    getch();
    return 0;
}
```

Assignment

- Write a program to input data of 10 students(roll, name, marks) structures. Write input data to the file "Student.txt" and write another program to read data from "Student.txt", and display data on console.

Writing Data to a Binary File

```
#include<iostream>
using namespace std;
#include<conio.h>
#include<fstream>
struct student
{
    int rollno;
    char name[30];
    int marks;
};
int main ()
{
    student s;int i;
    ofstream ofs("student.bin", ios_base::app);
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    cout <<"Give student data" << endl;
    cin>>s.rollno>>s.name>>s.marks;
    ofs.write(reinterpret_cast<char*>(&s),sizeof(s));
    ofs.close();
    getch ();
    return 0;
}
```

Reading Data from a Binary File

```
#include<iostream>
using namespace std;
#include<conio.h>
#include<fstream>
struct student
{
    int rollno;
    char name[30];
    int marks;
};
int main ()
{
    student s;
    ifstream ifs("student.bin");
    while(ifs.read(reinterpret_cast<char*>(&s),sizeof(s)))
  {
            cout<<"rollno = "<<s.rollno<<endl;
            cout<<"name = "<<s.name<<endl;
            cout<<"marks = "<<s.marks<<endl;
            cout<<endl;
  }
    ifs.close();
    getch ();
    return 0;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
}
```

Assignment

- Write a program to input data of 10 student (roll, name, marks) structures. Write the input data to a binary file “Student.bin”. Write another program to read data from the binary file and display data of the students having failed the exam (i.e. marks <50).
- Write a program to input data of 10 Product (PId, Pname, Qty) structures. Write data to a binary file “Product.bin”, then read data from the file and display only those products whose name starts with ‘A’.
- Write a program to input data of 10 Product (PId, Pname, Qty) structures. Write data to a binary file “Product.bin”, then read data from the file and display data of the products having product id in the range of 50 to 100.

Menus in C++ (Console-Based)

```
#include<iostream>

using namespace std;

#include<conio.h>

#include<fstream>

void add();

void multiply();

void div();

void subtract();

int main ()

{

    char op;

    while(1)

    {

        cout<<"\tENTER AN OPERATION"<<endl;

        cout<<"\tMENU"<<endl;

        cout<<"\t+.ADD NUMBERS"<<endl;

        cout<<"\t*.MULTIPLY NUMBERS"<<endl;

        cout<<"\t-.SUBTRACT NUMBERS"<<endl;

        cout<<"\t/.DIVISION OF NUMBERS"<<endl;

        cout<<"\te.Exit"<<endl;

        op=getche();

        cout<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
        if(op=='+')
        {
          add();
        }
        else if(op=='*')
        {
              multiply();
        }
        else if(op=='-')
        {
           subtract();
        }
        else if(op=='/')
        {
           div();
        } else if (op == 'e')
              break;
        else
        {
              cout<<"INVALID OPERATION"<<endl;
        }
    }
    getch ();
    return 0;
}
void add()
{
    int a,b,c;
    cout<<"enter 1st number"<<endl;
    cin>>a;
    cout<<"enter 2nd number"<<endl;
    cin>>b;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    c=a+b;
    cout<<"RESULT = "<<c<<endl;
}
void multiply()
{
    int a,b,c;
    cout<<"enter 1st number"<<endl;
    cin>>a;
    cout<<"enter 2nd number"<<endl;
    cin>>b;
    c=a*b;
    cout<<"RESULT = "<<c<<endl;
}
void subtract()
{
    int a,b,c;
    cout<<"enter 1st number"<<endl;
    cin>>a;
    cout<<"enter 2nd number"<<endl;
    cin>>b;
    c=a-b;
    cout<<"RESULT = "<<c<<endl;
}
void div()
{
    float a,b,c;
    cout<<"enter 1st number"<<endl;
    cin>>a;
    cout<<"enter 2nd number"<<endl;
    cin>>b;
    c=a/b;
    cout<<"RESULT = "<<c<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
}
```

File Handling Projects and Menus

Implement the following menu with file handling for the Student (rollno, name, marks) scenario:

1. Add Student data
2. Display Student data
3. Modify Student data
4. Delete Student data
5. Display student data with highest marks
6. Search a Student
7. Exit

```
#include<iostream>
#include<conio.h>
#include<fstream>
#include<string.h>
using namespace std;
struct student
{
    int rollno;
    char name[30];
    int marks;
};
void AddData();
void DisplayData();
void ModifyData();
void DeleteData();
void DisHighest();
void search();
int main()
{
    while(1)
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    {
        system("cls");
        char ch;
        cout<<"\n1.Add Student Data"<<endl;
        cout<<"2.Display All Data"<<endl;
        cout<<"3.Modify Student Data"<<endl;
        cout<<"4.Delete Student Data"<<endl;
        cout<<"5.Display Highest Marks"<<endl;
        cout<<"6.Search Record"<<endl;
        cout<<"7.Exit"<<endl;
        ch=getche();
        if(ch=='1')
            AddData();
        else if(ch=='2')
            DisplayData();
        else if(ch=='3')
            ModifyData();
        else if(ch=='4')
            DeleteData();
        else if(ch=='5')
            DisHighest();
        else if(ch=='6')
            search();
        else if(ch=='7')
            break;
        else
        cout<<"Ooooops!!!!!Invalid option!!!!!!!!"<<endl;
    }
    getch();
    return 0;
}
void AddData()
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
{
    student s;
    ofstream ofs("student.bin",ios_base::app);
    cout<<"Enter Student Roll No"<<"\t";
    cin>>s.rollno;
    cout<<"Enter Student Name"<<"\t";
    cin>>s.name;
    cout<<"Enter Student Marks"<<"\t";
    cin>>s.marks;
    ofs.write(reinterpret_cast<char*>(&s),sizeof(s));
    ofs.close();
}
void DisplayData()
{
    student s;
    ifstream ifs("student.bin");
    cout<<"\nRollno\t\t\tName\t\t\tMarks"<<endl;
    while(ifs.read(reinterpret_cast<char*>(&s),sizeof(s)))
    {
        cout<<s.rollno<<"\t\t\t"<<s.name<<"\t\t\t"<<s.marks<<endl;
    }
    ifs.close();
    getch();
}
void ModifyData()
{
    student s;
    int r;
    cout<<"Enter Rollno of the student whose data you want to modify"<<"\t";
    cin>>r;
    ifstream ifs("student.bin");
    ofstream ofs("temp.bin");
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
        while(ifs.read(reinterpret_cast<char*>(&s),sizeof(s)))
        {
            if(s.rollno==r)
            {
                cout<<"Enter New Data of student"<<endl;
                cout<<"Enter New Name\t";
                cin>>s.name;
                cout<<"Enter New Marks\t";
                cin>>s.marks;

            }
            ofs.write(reinterpret_cast<char*>(&s),sizeof(s));
        }
        ifs.close();
        ofs.close();
        //Copy back from temp.bin to student.bin
        ifstream ifs2("temp.bin");
        ofstream ofs2("student.bin");
        while(ifs2.read(reinterpret_cast<char*>(&s),sizeof(s)))
        {
            ofs2.write(reinterpret_cast<char*>(&s),sizeof(s));
        }
        ifs2.close();
        ofs2.close();
    }
    void DeleteData()
    {
        student s;
        int r,n=0;
        cout<<"Enter Rollno of the student whom you want to delete"<<endl;
        cin>>r;
        ifstream ifs("student.bin");
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    ofstream ofs("temp.bin");
    while(ifs.read(reinterpret_cast<char*>(&s),sizeof(s)))
    {
        if(s.rollno!=r)
        {
            ofs.write(reinterpret_cast<char*>(&s),sizeof(s));
            n++;
        }

    }
    ifs.close();
    ofs.close();
    if(n!=0)
    {

        ifstream ifs2("temp.bin");
        ofstream ofs2("student.bin");
        while(ifs2.read(reinterpret_cast<char*>(&s),sizeof(s)))
        {
            ofs2.write(reinterpret_cast<char*>(&s),sizeof(s));
        }
        ifs2.close();
        ofs2.close();
    }
    else
        cout<<"Rollno you entered does not exist";
    getch();
}
void DisHighest()
{
    student s;
    int count=0;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    ifstream ifs("student.bin");
    while(ifs.read(reinterpret_cast<char*>(&s),sizeof(s)))
    {
        count++;
    }
    ifs.close();
    student s2[count];int i=0;
    ifstream ifs2("student.bin");
    while(ifs2.read(reinterpret_cast<char*>(&s2[i]),sizeof(s2[i])))
    {
        i++;
    }
    ifs2.close();
    int large,loc;
    large=s2[0].marks;
    loc=0;
    for(i=1;i<count;i++)
    {
        if(s2[i].marks>large)
        {
            large=s2[i].marks;
            loc=i;
        }
    }
    cout<<"\nRollno\t\t\tName\t\t\tMarks"<<endl;
    cout<<s2[loc].rollno<<"\t\t\t"<<s2[loc].name<<"\t\t\t"<<s2[loc].marks<<endl;
    getch();
}
void search()
{
    student s;
    int r;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
        cout<<"Enter Rollno for data search"<<" ";
        cin>>r;
        ifstream ifs("student.bin");
        while(ifs.read(reinterpret_cast<char*>(&s),sizeof(s)))
        {
            if(s.rollno==r)
            {
                cout<<"\nRollno\t\t\tName\t\t\tMarks"<<endl;
                cout<<s.rollno<<"\t\t\t"<<s.name<<"\t\t\t"<<s.marks<<endl;
            }
        }
        ifs.close();

}
```

Presenting SubMenus to Users

```
#include<iostream>
using namespace std;
#include<conio.h>
#include<fstream>
struct student
{
    int rollno;
    char name[30];
    int marks;
};
struct teacher
{
    int tid;
    char name[30];
};
void addStudent()
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
{
    student s;
    cout<<"ENTER STUDENT DATA"<<endl;
    cin>>s.rollno;
    cin>>s.name;
    cin>>s.marks;
    ofstream ofs("STUDENT.BIN",ios_base::app);
    ofs.write(reinterpret_cast<char*>(&s),sizeof(s));
    ofs.close();
}
void addTeacher()
{
    teacher t;
    cout<<"ENTER DATA OF TEACHER"<<endl;
    cin>>t.Tid;
    cin>>t.name;
    ofstream ofs("TEACHER.BIN",ios_base::app);
    ofs.write(reinterpret_cast<char*>(&t),sizeof(t));
    ofs.close();
}
void addData()
{
    while(1)
    {
        char ch;
        system("cls");
        cout<<"ENTER AN OPTION FROM THE MENU"<<endl;
        cout<<"1.ADD Student Data"<<endl;
        cout<<"2.ADD Teacher Data"<<endl;
        cout<<"3.EXIT FROM THE SUB MENU"<<endl;
        ch=getche();
        cout<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
            if(ch=='1')
            {
                addStudent();
            }
             else if(ch=='2')
            {
                addTeacher();
            }
            else if(ch=='3')
            {
                break;
            }
            else
            {
                cout<<"INVALID OPTION"<<endl;
            }
        }
        getch ();
}
void displayData()
{
        ifstream ifs("STUDENT.BIN");
        student s;
        while(ifs.read(reinterpret_cast<char*>(&s),sizeof(s)))
        {
        cout<<"***** THE STUDENT DATA *****"<<endl;
        cout<<s.rollno<<endl;
        cout<<s.name<<endl;
        cout<<s.marks<<endl;
        cout<<"****************************"<<endl;
}
ifs.close();
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
getch ();
}
void modifyData()
{
    student s;
    int r;
    cout<<"GIVE STUDENT ROLLNO WHOSE DATA IS TO BE MODIFIED"<<endl;
    cin>>r;
    ifstream ifs("STUDENT.BIN");
    ofstream ofs("TEMP.BIN");
    while(ifs.read(reinterpret_cast<char*>(&s),sizeof(s)))
    {
        if(s.rollno==r)
        {
            cout<<"GIVE NEW ROLLNO,NAME AND MARKS"<<endl;
            cin>>s.rollno;
            cin>>s.name;
            cin>>s.marks;
        }
        ofs.write(reinterpret_cast<char*>(&s),sizeof(s));
    }
    ifs.close();
    ofs.close();
    ofstream ofs2("STUDENT.BIN");
    ifstream ifs2("TEMP.BIN");
    while(ifs2.read(reinterpret_cast<char*>(&s),sizeof(s)))
    {
        ofs2.write(reinterpret_cast<char*>(&s),sizeof(s));

    }
    ifs2.close();
    ofs2.close();
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    cout<<"DATA SUCCESSFULLY MODIFIED"<<endl;
    getch();
}
void deleteData()
{
    student s;int r;
    cout<<"GIVE ROLLNO WHOSE DATA IS TO BE DELETED"<<endl;
    cin>>r;
    ifstream ifs("STUDENT.BIN");
    ofstream ofs("TEMP.BIN");
    while(ifs.read(reinterpret_cast<char*>(&s),sizeof(s)))
    {
        if(s.rollno!=r)
        {
            ofs.write(reinterpret_cast<char*>(&s),sizeof(s));
        }
    }
    ifs.close();
    ofs.close();
    ifstream ifs2("TEMP.BIN");
    ofstream ofs2("STUDENT.BIN");
    while(ifs2.read(reinterpret_cast<char*>(&s),sizeof(s)))
    {
        ofs2.write(reinterpret_cast<char*>(&s),sizeof(s));
    }
    ifs2.close();
    ofs2.close();
    cout<<"DATA SUCCESSFULLY DELETED"<<endl;
}
void dispHigh()
{
    int count=0;student s;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    ifstream ifs("STUDENT.BIN");
    while(ifs.read(reinterpret_cast<char*>(&s),sizeof(s)))
    {
        count++;
    }
    ifs.close();
    student s2[count];int i=0;
    ifstream ifs2("STUDENT.BIN");
    while(ifs2.read(reinterpret_cast<char*>(&s2[i]),sizeof(s2[i])))
    {
        i++;
    }
    ifs2.close();

    int large=s2[0].marks;
    int loc=0;
    for(i=0;i<=count;i++)
        if(s2[i].marks>large)
        {
            large=s2[i].marks;
            loc=i;
        }
    cout<<"*****STUDENT HAVING HIGHEST MARKS*****"<<endl;
    cout<<"ROLLNO = "<<s2[loc].rollno<<endl;
    cout<<"NAME = "<<s2[loc].name<<endl;
    cout<<"MARKS = "<<s2[loc].marks<<endl;
    getch ();
}
void search ()
{
    student s;int r;
    cout<<"ENTER A NUMBER WHICH YOU WANT TO SEARCH "<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
    cin>>r;
    ifstream ifs("STUDENT.BIN");
    while(ifs.read(reinterpret_cast<char*>(&s),sizeof(s)))
    {

        if(s.rollno==r)
       {
            cout<<endl;
            cout<<"\t rollno =  "<<s.rollno<<endl;
            cout<<"\t name  = "<<s.name<<endl;
            cout<<"\t marks = "<<s.marks<<endl;
      }
}
    ifs.close();
    cout<<"PRESS ANY KEY TO CONTINUE"<<endl;
    getch();
}



int main ()
{
    char ch ;
    while(1)
    {  system("cls");
       cout<<"MAIN MENU"<<endl;
        cout<<"1.ADD DATA"<<endl;
        cout<<"2.DISPLAY STUDENT DATA"<<endl;
        cout<<"3.MODIFY STUDENT DATA"<<endl;
        cout<<"4.DELETE STUDENT DATA"<<endl;
        cout<<"5.DISPLAY STUDENT DATA WITH HIGHEST MARKS"<<endl;
        cout<<"6.SEARCH STUDENT DATA"<<endl;
        cout<<"7.EXIT STUDENT DATA"<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
if(ch=='1')
{
    addData();
}
else if(ch=='2')
{
    displayData();
}
else if(ch=='3')
{
    modifyData();
}
else if(ch=='4')
{
    deleteData();
}
else if(ch=='5')
{
   dispHigh();
}
else if(ch=='6')
{
    search();
}
else if(ch=='7')
{
    break;
}
else
{
    cout<<"INVALID OPTION"<<endl;
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
            }
        }
        cout<<"PRESS ANY KEY TO CONTINUE"<<endl;
        getch ();
        return 0;
    }
```

Projects with Menus and SubMenus

Purchase and Sale Management System

Product (PId,Pname,Balance)

Purchase(prid, pid ,qty, ppu)

Sales(salid, pid, qty, ppu)

Product

Add new product

Display product

Modify product

Delete a product

Purchase

Purchase a product

Display all purchases

Sales

Sell a product

Display sale transaction

Display all sales

Display Revenue

Rules

- A product with a pid should be unique
- We can purchase or sell a product only if it exists in product file.
- We can not sell a product more than its balance in product file.

Examination System

class (classId, classname)

subject(subjectId, subject name)

result(classId, rollno, subjectId, marks)

student(classId, rollno, name)

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Class

Add class

Display classes

Modify classes

Delete class

Student

Add student

Search a student

By rollno
By name

Modify student

Delete student

Result

Add result

Search result

By class
By rollno

Subject

Add Subject

Display Subjects

Modify subject

Delete subject

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

PYTHON PROGRAMMING FUNDAMENTALS

Python is a high-level, dynamic, free and open-source language in which we can easily develop programs . In contrast to the compiled languages such as C++, it is an interpreted language, Its supports object-oriented programming, and provides support of many libraries which are widely used to develop programs from simple applications to complex artificial intelligence based applications.

The Python blocks are formed using indentation.

Example

```
print("first")
```

```
I = 2
if I == 2:
  print (“yes”)
else:
  print (“No”)
```

Example

```
if I ==2:
      print (“one”)
elif I ==2:
      print (“Two”)
elif I== 3 :
      print (“Three”)
else:
      print (“Invalid”)
```

Input-Output in Python

```
a =  (int) (input(“Give value”))
print (“a=”,a)
```

Write a program in Python to input m & n, and display the numbers from n to m.

```
n=(int) (input("Give Value of N"))
m=(int) (input("Give Value of M"))
for i in range(n,m+1):
      print(i)
```

Python Operators

+, -, ~ Unary

** power

*, /, %

+, -

<<, >>

&

^ exclusive or

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

| or

Not, and, or

Relational Operators

<, >, <=, >=, ==, !=

Loops in Python

For-Loop

```
for i in range ( 1, 11):
        print (i)
print ("yes")
```

Assignments

a. Write Python code to print even numbers from 2 to 100.

b. Write Python code to input a number and display whether it is even or odd.

Write Python code to input a number n and display sum of the series:

1+2+3+....+n

```
n = (int)(input("Give n"))
i = 1
Sum = 0
for i in range ( 1, n+1):
    Sum = Sum +i
print (Sum)
```

Write Python code to input x and n and display sum of following series

(x+1)+(x+2)+....+(x+n)

```
N = (int) (input("Give N"))
X = (int) (input("Give x"))
Sum = 0
I = 1
for I in range (1, N+1):
    Sum = Sum + (X+I)
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
print (Sum)
```

Write python code to print sum of the following series using for loop and while loop:

(1-2)-(2+3)+....±(n±(n±1))

```
n=(int)(input("Give value of n="))
i=1
sum=0
sign=(-1)
sign2=1
for i in range(1,n+1):
    sum=sum+(i+(i+1)*sign)*sign2
    sign=sign*(-1)
    sign2=sign2*(-1)
print("Sum using for-loop is:",sum)
i=1
sum=0
sign=(-1)
sign2=1
while i<=n:
    sum=sum+(i+(i+1)*sign)*sign2
    sign=sign*(-1)
     sign2=sign2*(-1)
     i=i+1
print("Sum using while-Loop is:",sum)
```

Data Structures in Python

The data structure existing in Python are lists, tuples, range, dictionary.

Lists

Lists are mutable sequences of objects indexed by natural number that can be changed after creation. The lists are enclosed in brackets.

```
Lst = [1, 2, 3, 4]
Lst = [1, “abc”, 2]
```

Tuples

A tuple is a sequence of immutable Python objects. The general format is

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Variable = (element 1, element 2, ……..)

e.g

t = (1,2,3)

It is similar to a list, however it is immutable.

Ranges

A range is a list of integers. There is a built in function range() for this data structure.

R = range (2,5)

print (R)

R = range(2, 10, 2)

print (R)

Dictionaries

Dictionaries in Python are the structures used for mappings. The keys are mapped to values.

Variable = {key1: value1, key2: value2, ….}

d = {“1”: “a”, “2”: “b”, “3”: “c”}

print (d[“1”])

Programs using List Processing in Python

Write Python code to input a list of n elements and display the highest value with its location.

```
list1=[]
n=int(input("Number of Elements of list:"))

for i in range(0, n):

   print("Enter ",i+1 , "th element: ", end='')

   list1.append(element)


print("Your list is as follows: ", list1)

high = list1[0]

loc = 0

for i in range(0, n):

    if list1[i] > high:

        high = list1[i]

        loc = i

print("Highest element is ", high, "at index", loc, ".")
```

Prof. Minhaj Ahmad Khan Lecture Notes by Minhaj Ahmad Khan

Two Dimensional Lists / Arrays

list1 = [[]]

list2 = [[1,2,3],[4,5,6]]

For I in list2:

print(i)

for I in range (0, len(list2)):

for j in range (0, len(list2[i]):

print (list2[i][j])

Assignment

Write python code to input 2-dim list of order 3*3 and display the input elements.

Write python code to input 2-dim list of 3*3 and display the

- Highest number row wise
- Highest number column wise
- Sum of each diagonal

Code

Notes by Minhaj Ahmad Khan

```
list1 = [[0, 0, 0],
         [0, 0, 0],
         [0, 0, 0]]

      for r in range(0, 3):
    for c in range(0, 3):
        print("Enter", "[", r, "]", "[", c, "] element of matrix:", end=' ')
        list1[r][c] = int(input())
print(list1)

print("----------For Rows----------")
row_high = 0
for r in range (0, 3):
    row_high = list1[r][0]
    for c in range (0, 3):
        if list1[r][c] > row_high:
            row_high = list1[r][c]
    print("Highest in", r + 1, "th row is : ", row_high)

print("----------For Columns----------")
col_high = 0
for c in range (0, 3):
    col_high = list1[0][c]
    for r in range (0, 3):
        if list1[r][c] > col_high:
            col_high = list1[r][c]
    print("Highest in", c + 1, "th column is : ", col_high)

print("----------For Primary Diagonal----------")

sum = 0
for r in range (0, 3):
```

```
    for c in range (0, 3):
        if (r == c) :
            sum = sum + list1[r][c]

print("The sum of primary diagonal of matrix is :", sum)
print("----------For Secondary Diagonal----------")
sum2 = 0
for r in range (0, 3):
    sum2 = sum2 + list1[r][3-r-1]

print("The sum of secondary diagonal of matrix is :", sum2)
```

List of Records /Structures

Write python code to input 10 student (rollno, name, marks) records and display the student with the highest marks.

Code

```
list1 = [[0, " ", 0] * 10]

for i in range (0, 10):
    print(i+1, "th student RollNo: ", end='')
    list1[i][0] = int(input())
    print(i+1,"th student Name: ",end='')
    list1[i][1] = input()
    print(i+1, "th student Marks: ", end='')
    list1[i][2] =int(input())

loc = 0
high = list1[0][2]
for i in range (0, 10):
    if list1[i][2]> high :
       high = list1[i][2]
       loc = i


print("The student with the highest marks is: ")
print("Roll No:",list1[loc][0] )
print("Name:",list1[loc][1])
print("Marks:",list1[loc][2])
```

Assignment

Write python code to input 10 product (pid, pname, qty) records using lists and display the product with lowest quantity.

Accessing SubLists

List2=[1,2,3,4,5,6,7,8,9,10]

List2[0]=5

List2.insert(5,40)

print(list2)

```
print(list2[1:4])
print(list2[:])
print(list2[1:])
print(list2[:2])
print(list2[:-2])
print(list[-2:])
print(len(list2))
```

File Handling in Python

```
############### File handling in Python
        f1 = open("file1.txt", "wt")
        f1.write("1st line\n")
        f1.write("second line\n")
        f1.close()
        ######### Read data from file
        f2 = open("file1.txt", "rt")
        l1=f2.readline()
        while l1:
            print(l1,end='')
            l1 = f2.readline()
        f2.close()
        ############ Read data from file
        f3 = open("file1.txt", "rt")
        for l1 in f3.readlines():
            print(l1)
        f3.close()
##################
```

Write program to input Student(rollno, name, marks) data, write to a a text file "student.txt". Read data from the file and display it.

```
        f1=open("student.txt", "at")
        r = (int)(input("give rollno"))
```

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

```
n=input("give name")
m=(int)(input("give marks"))
f1.write(str(r) +"\n")
f1.write(n + "\n")
f1.write(str(m)+"\n")
f1.close()
f2 = open("student.txt", "rt")
l1=f2.readline()
while l1:
    print(l1)
    1 = f2.readline()
f2.close()
```

Assignment

Write a program in Python to input Product(PId, PName, Qty) data, and store it to a file "product.txt". Write another program to read data from the file and display data of the product having the highest quantity**.**

Functions in Python

```
def sum1(a,b):
  return a+b
k = sum1( 5,6 )
Print ("k = ", k)
```

Recursion

```
def factorial (n):
  if n==0:
      return 1
  else:
      return n * factorial ( n-1)
```

Modules, Packages and Namespaces in Python

A module is a collection of classes functions and variables. A package is a collection.

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

The modules can be imported using import statement. e.g.

```
import sys
 if k==6:
   sys.exit()
import os
print (os.name)
```

Namespaces in Python

Python uses dynamic namespaces. Each function module, and class define its own namespace. Python uses following from namespace.

Built names…..int string, def etc.

Global…….declared as global at the top level.

Local…….assigned inside a function.

Using Exception Handling

The try block is used to contain statements due to which an exception may occur. The "except" block contains the exception handler which is executed in case an error occurs. The "else" block is executed if no error occurs. The "finally" block is executed whether or not the error occurs.

```
try:
    print (“ok”)
except:
    print (“Exception handler”)
else:
    print("Else block in Exception")
```

Project with Menu in Python

Implement the following project with file handling menu in Python:

Student

Add student

Search a student

By rollno

By name

Modify student

Delete student

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan

Result

- **Add result**
- **Search result**
  - **By class**
  - **By rollno**

Class

- **Add Class**
- **Display Class**
- **Modify Class**
- **Delete**

Programming Fundamentals Lecture Notes by Minhaj Ahmad Khan