**What is File Handling is C++?**

File handling in C++ is a mechanism to store the output of a program in a file and help perform various operations on it. Files help store these data permanently on a storage device.

Or we can say, File handling is a process of reading and writing data into the files

**Main File Handling operations:**

* Creation of a file.
* open a file
* read a file
* write data
* copy file
* Delete a file

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

/\**When we want to store our data in secondary memory, we need file instead*

*of variables. In order to work with files we need a header file i.e.<fstream>.\*/*

**How to create a file?**

ofstream onfile;

onfile.open("Alpha.txt");

cout<<"File created successfully"<<endl;

onfile.close();

//Here we can create different types of files using their respective extensions.

**How to write in the file?**

ofstream onfile;

onfile.open("Alpha.txt");

onfile<<"We are Indian. India is a great nation."<<endl;

cout<<"Data has been written successfully"<<endl;

onfile.close();

**How to read a file?**

ifstream infile;

string mydata;

infile.open("Alpha.txt");

infile>>mydata;

cout<<mydata;

infile.close();

Output:In this way, we don't get complete data.

**How to get complete file data?**

ifstream infile;

string mydata;

infile.open("Alpha.txt");

while(getline(infile,mydata))

{

cout<<mydata;

}

infile.close();

**How to copy data from one file to another file?**

ifstream infile;

ofstream onfile;

char mydata;

infile.open("Alpha.txt");

onfile.open("Beta.txt");

while(infile.get(mydata))

{

onfile.put(mydata);

}

cout<<"Data copied!";

infile.close();

onfile.close();

**How to delete a file?**

int value=remove("Beta.txt");

if(value==0)

{

cout<<"File is deleted successfully:"<<endl;

}

else

{

cout<<"File is not deleted successfully:"<<endl;

}

}

**What is Exception Handling?**

A condition that occurs during the execution of a program is referred to as an exception.

And exception handling in C++ is an answer to a rare occurrence that occurs during the

execution of a program

In C++, we have try-catch block to handle the exceptions.

try

{

Write your risky code here;

}

catch(Exception)

{

Write the code to handle the exception;

}

**Simple example of exception handling**

#include<iostream>

#include<stdexcept>

using namespace std;

int main()

{

int a,b,c;

try

{

cout<<"Enter two numbers:"<<endl;

cin>>a>>b;

if(a==0 || b==0)

{

cout<<"Numbers must be non-zero:"<<endl;

}

c=a/b;

cout<<"Division of numbers is="<<c<<endl;

}

catch(exception e)

{

cout<<"Division by zero is not allowed"<<endl;

}

}

**Another example of Exception Handling.**

#include<iostream>

#include<stdexcept>

using namespace std;

int main()

{

double bal=1000.00;

double amt;

try

{ //Desposit

cout<<"Enter deposit amount:";

cin>>amt;

if(amt<=0)

{

cout<<"Invalid deposit amount:"<<endl;

}

bal=bal+amt;

cout<<"Total available balance is:"<<bal<<endl;;

//Withdraw

cout<<"Enter withdrawn amount:";

cin>>amt;

if(amt<=0)

{

cout<<"Invalid withdrawal amount:"<<endl;

}

if(amt>bal)

{

cout<<"Insufficient balance:"<<endl;

}

bal=bal-amt;

cout<<"Total available balance is:"<<bal;

}

catch(exception e)

{

cout<<"e.what()";

}

}