Object Oriented Programming

Exception handling & File handling

Exception handing in C++

- Exceptions are run-time anomalies or abnormal conditions that a program encounters during its execution. C++ provides following specialized keywords for this purpose.
- *try*: represents a block of code that can throw an exception.
- *catch*: represents a block of code that is executed when a particular exception is thrown.
- *throw*: Used to throw an exception. Also used to list the exceptions that a function throws, but doesn’t handle itself.

Divided by zero exception

```
int main()
{
float x, y;
cout << "enter numerator \n";
cin>>x;
cout << "enter denominator \n";
cin>>y;
cout<<x/y;

return 0;
}
```

```
int main()
{
int x = -1;
cout << "Before try \n";
try {
        cout << "Inside try \n";
                throw x;
        cout << "After throw (Never executed) \n";

} catch (int x ) {
        cout << "Exception Caught \n";
}
cout << "After catch (Will be executed) \n";
return 0;
}
```

- Output

Before try

Inside try

Exception Caught

After catch (Will be executed)

```
#include <iostream>
using namespace std;
double division(int a, int b) {
  if( b == 0 ) {
    throw "Division by zero condition!";
  }  return (a/b);
}
int main () {
  int x = 50;    int y = 0;    double z = 0;
   try {
    z = division(x, y);
    cout << z << endl;
  } catch (const char* msg) {
    cout << msg << endl;   } }
```

Catch Block

- There is a special catch block catch(…) that can be used to catch all types of exceptions. For example, in the following program, an int is thrown as an exception, but there is no catch block for int, so catch(…) block will be executed.

```
int main()
{
        try {
        throw 10;
        }
        catch (char excp) {
                cout << "Caught " << excp;
        }
        catch (...) {
                cout << "Default Exception\n";
        }
        return 0;
}
```

- Output

Default Exception

File Handling in C++

File Handling

- In C++, files are mainly dealt by using three classes fstream, ifstream, ofstream available in fstream headerfile.
  ofstream: Stream class to write on files
  ifstream: Stream class to read from files
  fstream: Stream class to both read and write from/to files.

- The first step is to open the particular file for read or write operation. We can open file by passing file name in the open method.

- When an object is created for ofstream class, it allows us to write into a file just like cout. When opening a file with ofstream object if file is present then the content is written else it is created.
- When an object is created for ifstream class, it allows us to input from a file just like cin. getline takes the entire line at once.

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
ofstream ofile;
ofile.open ("text.txt");
ofile << “Hello world!" << endl;
cout << "Data written to file" << endl;
ofile.close();
return 0;
}
```

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{	char data[100];
	ifstream ifile;
	ifile.open ("text.txt");
	while ( !ifile.eof() )
	{
		ifile.getline (data, 100);
		cout << data << endl;
	}
	ifile.close();
	return 0;
}
```

Lab work

- Create a file in C drive for writing your name, Id, and study program. Read the file and display the data written in the file on screen.

References

- https://www.geeksforgeeks.org/file-handling-c-classes/
- https://www.geeksforgeeks.org/output-c-programs-set-34-file-handling/
- https://www.geeksforgeeks.org/exception-handling-c/