Lab's Scope:

- **File Handling (Read/Write) – Part 2**
- **Exception Handling**

Problem 1:

Write a program that has an array of Students where each student is a variable from a `struct` having members name, ID, and grade. Your program should create an array of 3 students and write their information in a file. Then the program should read the file, calculate the average grade of the students, and append the average to the end of the file.

Solution:

```
using namespace std;
#include <iostream>
#include <fstream>

struct Student
{
	string name;
	int ID;
	float grade;
};
const int SIZE = 3;

int main()
{
	Student arr[SIZE] = {{"Ahmed",1234,3.5}, { "Omar",2358,2.5},
{"Mariam",2888,4.0}};
	float sum = 0, average;
	ofstream fout;
	fout.open("List.txt");
	for (int i = 0; i < SIZE; i++)
	{
		fout << arr[i].name<<"\t"<<arr[i].ID<<"\t"<<arr[i].grade<<endl;
	}
	fout.close();
	ifstream fin;
	fin.open("List.txt",ios::in);
	for (int i = 0; i < SIZE; i++)
	{
		fin>> arr[i].name>>arr[i].ID>>arr[i].grade;
		sum += arr[i].grade;
		cout << sum << endl;
	}
	average = sum / SIZE;
	fin.close();

	fout.open("List.txt",ios::app);
	fout << average;
	fout.close();
	return 0;
}
```

Problem 2:

Update the program from problem 1, so that your program has three functions instead of writing the whole code in the main application, as follows:

- A function called “writeToFile” for writing the student information.
- A function called “readFromFile”, for reading the student information, and calculating the sum and average.
- A function called “appendToFile” that appends the average at the end of the file.

Hint: think carefully about your parameters and the return type.

Solution:

```
using namespace std;
#include <iostream>
#include <fstream>

struct Student
{
	string name;
	int ID;
	float grade;
};
const int SIZE = 3;
//////////////////////////////////////////////////////////////////////////
void writeToFile(ofstream& fout, Student arr[])
{
	fout.open("List.txt");
	for (int i = 0; i < SIZE; i++)
	{
		fout << arr[i].name << "\t" << arr[i].ID << "\t" << arr[i].grade <<
endl;
	}
	fout.close();
}
//////////////////////////////////////////////////////////////////////////
float readFromFile(ifstream& fin, Student arr[])
{
	float sum = 0, average;
	fin.open("List.txt", ios::in);
	for (int i = 0; i < SIZE; i++)
	{
		fin >> arr[i].name >> arr[i].ID >> arr[i].grade;
		sum += arr[i].grade;
		cout << sum << endl;
	}
	average = sum / SIZE;
	fin.close();
	return average;
}
//////////////////////////////////////////////////////////////////////////////
```

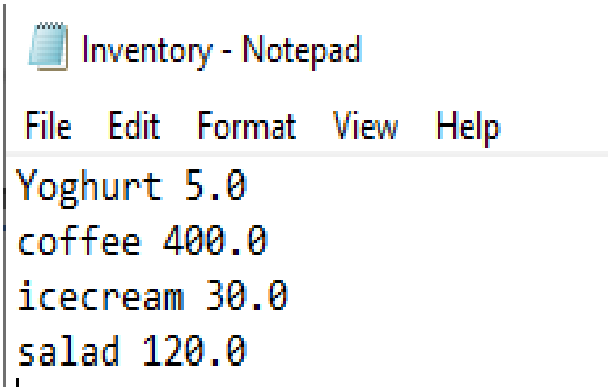
```
void appendToFile(ofstream& fout, float average)
{
	fout.open("List.txt", ios::app);
	fout << average;
	fout.close();
}
//////////////////////////////////////////////

int main()
{
	Student arr[SIZE] = {{"Ahmed",1234,3.5}, { "Omar",2358,2.5},
{"Mariam",2888,4.0}};

	ofstream fout;
	ifstream fin;

	writeToFile(fout, arr);
	float avg = readFromFile(fin, arr);
	appendToFile(fout, avg);

	return 0;
}
```

Problem 3:

A supermarket has its products and their prices stored in a file (the file is available on e-learning under the name “Inventory.txt”). The program should do the following:

- Read the file and display each product to the user.
- The user should state how many items he needs from the product.
- The program should then calculate the total bill and write the items and the total bill in a file called “Bill.txt”

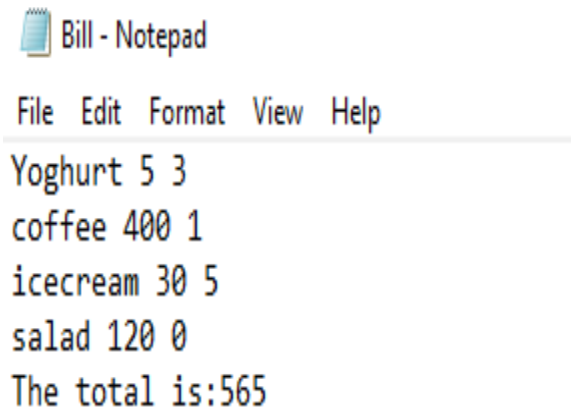
Screenshot of Inventory.txt file



Sample Execution:

```
For each product, select how many items:
Yoghurt
3
Coffee
1
Icecream
5
salad
0
The total payment is 565.0
```

Screenshot of the output Bill.txt file



Solution:

```
using namespace std;
#include <iostream>
#include <fstream>

int main()
{
	ofstream fout;
	ifstream fin;
	string product;
	float price, total =0;
	int items;
	fin.open("Inventory.txt");
	fout.open("Bill.txt");
	cout << "For each product, select how many items:" << endl;

	while (!fin.eof())
	{
		fin >> product;
		fin >> price;
		cout<< product<<endl;
		cin >> items;
		fout << product << " " << price << " " << items << endl;
		total += (items * price);
	}
	 cout << "The total payment is " << total << endl;
	fout << "The total is:" << total << endl;
	fin.close();
	fout.close();

	return 0;
}
```

Problem 4:

Write a program for a cinema ticketing system where the program takes the user's age as input. The program should use Exception handling throwing an exception if the user's age is less than 12, printing out a message "Violent movies are not suitable for that age", otherwise the program should print "You are allowed to enter the movie."

Solution:

```
using namespace std;
#include <iostream>

int main()
{
	int age;
	cout << "Enter your age:";
	try
	{
		cin >> age;
		if (age < 12)
			throw age;
		cout << "You are allowed to enter the movie." << endl;
	}
	catch (int age)
	{
		cout << "Violence movies are not suitable for that age." << endl;
	}
	return 0;
}
```

Problem 5:

Write a program for the University where the program takes the student' ID as a string input. The program should use Exception handling throwing an exception if the first letter of the ID is not character '2'. The program should allow the user to enter his ID again if an exception occurs.

Solution:

```
using namespace std;
#include <iostream>
#include <string>
int main()
{
	string ID;
	cout << "Enter your ID:";
	try
	{
		cin >> ID;
		if (ID.at(0) != '2')
			throw ID;
		cout << "Correct ID" << endl;
	}
	catch (string ID)
	{
		cout << "Incorrect ID, Try again." << endl;
		cin >> ID;
	}
	return 0;
}
```

Problem 6:

Write a program that traverses and prints an array of integers of size 4. Write a for loop that traverses till the fifth element and add an Exception handling that throws -1 where your program should not crash when reaches the fifth element, however, it should print "Array out of boundaries" and should continue normally.

Solution:

```
using namespace std;
#include <iostream>
#include <string>

int main()
{
	int arr[4] = { 10,20,30,40 };
	try
	{
		for (int i = 0; i <= 4; i++)
		{
			if (i == 4)
				throw -1;

			cout << arr[i] << endl;
		}
	}
	catch (int x)
	{
		cout << "Array out of boundaries" << endl;

	}
	cout << "The program will continue normally." << endl;
	return 0;
}
```