|  |
| --- |
| **Computer Engineering Department - ITU** |
| **CE101L: Object Oriented Programming Lab** |

|  |  |
| --- | --- |
| **Course Instructor: Usama Bin Shakeel** | **Dated: 17/03/2022** |
| **Teaching Assistant: Aqsa Khalid** | **Semester: Spring 2022** |
| **Lab Engineer: Nadir Abbas** | **Batch: BSCE2021** |

# **Lab 2B. Reading Data and Writing Different Shapes using File Handling**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Roll number** | **Report**  **(out of 100)** | **Scaled to 10** | **Total**  **(out of 10)** |
| NIMRA MAQBOOL | BSCE21012 |  |  |  |

Checked on: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Objective**

The objective of this lab is to practice problems related to nested loops and switch cases and file handling.

## **Equipment and Component**

|  |  |  |
| --- | --- | --- |
| **Component Description** | **Value** | **Quantity** |
| Computer | Available in lab | 1 |

## **Conduct of Lab**

1. Students are required to perform this experiment individually.
2. In case the lab experiment is not understood, the students are advised to seek help from the course instructor, lab engineers, assigned teaching assistants (TA) and lab attendants.

## **Theory and Background**

A statement or a set of statements that is executed repeatedly is called a loop. The statement(s) in a loop are executed for a specified number of times or until some given condition remains true. In C++, there are three kinds of loop statements. These are: The “for” loop. The “while” loop. The “do-while” loop.

The “**while**” loop: - It is a conditional loop statement. It is used to execute a statement or a set of statements as long as the given condition remains true. In **for** loop the number of iterations to be done is already known. In other words, it is used to execute a statement or a set of statements for a fixed number of times.

A loop structure completely inside the body of another loop structure is called a **nested** loop. A nested loop has one loop inside of another. When a loop is nested inside another loop, the inner loop runs many times inside the outer loop. ... In each iteration of the outer loop, the inner loop will be re-started.

**File Handling:** In all programs we have written so far the input was given from the keyboard and output was received on the computer screen. This method works well when inputs are not too many and are small. But when the input becomes large it is not possible to enter data from the keyboard. If there is a mistake in entering data all data will have to be entered again. Similarly if the output is large it is necessary to keep it permanently stored on the disk for future use.

The data files provide a solution to this problem. The files stored on disk are used to give input into the program. The input data is written into the program. The program reads the data from the file. Similarly the program writes the output in a file on the disk. The output written in the file can then be displayed and/or printed on the printer. File Handling in C++ is quite different from that in procedural languages. C++ provides an easy way of File Handling. In procedural languages files are processed on the disk by using functions but in C++ objects are used for the same purpose.

**Streams and Files**

In C++ special classes known as stream classes are used to handle data streams. These classes are called stream classes. Stream classes are defined in header files and these are included in the program through their header files. The objects of stream classes are used to control flow of data.

For example, in C++ programs we commonly include the header file “iostream”. This header file contains the objects “cin”, “cout”, “cerr”, and “clog”. These objects are used to move streams of data across various parts of the computer. Following are the most commonly used C++ streams classes

**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.

**Lab Task**

**Task A**

Write a C++ code to output different shapes using the concept of nested

loops and file handling in multiple steps given below:

* Take input from a file
* Take input from a user
* Pass this information to a function to print shapes
* Write these shapes to a file

Calendar

Description automatically generated

|  |
| --- |
| void oddRightTriangle(int &num,fstream &myFile) {  myFile.open("shapesTask1.txt" , ios::out);  int j=0;  if(!myFile.is\_open()){  cout<<"the file you are trying to open is not open"<<endl;  myFile<<"the file you are trying to open is not open"<<endl;  }  else{  myFile<<"TASK 1 IS :"<<endl;  for(int i=1;i<=num;i++){  for(j=0;j<i;j++)  if(i%2==1){  cout<<i;  myFile<<i;  }  for(int k=0;k<=j;k++){  if(i%2==0){   cout<<"#";  myFile<<"#";  }  }   cout<<endl;  myFile<<endl;  }  }  myFile<<endl;  myFile.close();  } void oddRightTriangle1(int &rows,fstream &myFile,ifstream &files) {  myFile.open("shapesTask1.txt" , ios::app);  int j=0;  if(!myFile.is\_open()){  cout<<"the file you are trying to open is not open"<<endl;  myFile<<"the file you are trying to open is not open"<<endl;  }  else{  myFile<<"TASK 1 IS :"<<endl;  for(int i=1;i<=rows;i++){  for(j=0;j<i;j++)  if(i%2==1){  cout<<i;  myFile<<i;  }  for(int k=0;k<=j;k++){  if(i%2==0){   cout<<"#";  myFile<<"#";  }  }   cout<<endl;  myFile<<endl;  }  }  myFile<<endl;  myFile.close(); } void invertedRight(int &num,fstream &myFile){  myFile.open("shapesTask1.txt",ios::app);  if(!myFile.is\_open()){  cout<<"the file you are trying to open is not open"<<endl;  myFile<<"the file you are trying to open is not open"<<endl;  }  else{  myFile<<"TASK 2 IS :"<<endl;  for(int j=1;j<=num;j++){  for(int i=num;i>=j;i--){  if(j%3==1){  cout<<"#";  myFile<<"#";  }  if(j%3==2){  cout<<i;  myFile<<i;  }  if(j%3==0){  cout<<"$";  myFile<<"$";  }  }  cout<<endl;  myFile<<endl;  }  }  myFile<<endl;  myFile.close();  } void invertedRight1(int &rows,fstream &myFile,ifstream &files ){  myFile.open("shapesTask1.txt",ios::app);  if(!myFile.is\_open()){  cout<<"the file you are trying to open is not open"<<endl;  myFile<<"the file you are trying to open is not open"<<endl;  }  else{  myFile<<"TASK 2 IS :"<<endl;  for(int j=1;j<=rows;j++){  for(int i=rows;i>=j;i--){  if(j%3==1){  cout<<"#";  myFile<<"#";  }  if(j%3==2){  cout<<i;  myFile<<i;  }  if(j%3==0){  cout<<"$";  myFile<<"$";  }  }  cout<<endl;  myFile<<endl;  }  }  myFile<<endl;  myFile.close();  } void evenRightTriangle(int &num,fstream &myFile) {  int d;  int i;  int j;  int num1=2;  myFile.open("shapesTask1.txt",ios::app);  if(!myFile.is\_open()){  cout<<"the file you are trying to open is not open"<<endl;  myFile<<"the file you are trying to open is not open"<<endl;  }  else {  myFile<<"TASK 3 IS :"<<endl;  for (i = 1; i <= num; i++) {  if (i % 2 == 1) {  for (j = 1; j <= num - i; j++) {  cout << " ";  myFile<< " ";  }  d = num1;   for (j = 1; j <= i; j++) {  cout << num1;  myFile<<num1;  }  num1 += 2;  }  if (i % 2 == 0) {  for (j = 1; j <= num - i; j++) {  cout << " ";  myFile<< " ";  }  d = i;  for (j = 1; j <= i; j++) {  cout << "#";  myFile<< "#";  }  }   cout << endl;  myFile<< endl;  }  }  myFile<<endl;  myFile.close(); } void evenRightTriangle1(int &rows,fstream &myFile,ifstream &files) {  int d;  int i;  int j;  int num1=2;  myFile.open("shapesTask1.txt",ios::app);  if(!myFile.is\_open()){  cout<<"the file you are trying to open is not open"<<endl;  myFile<<"the file you are trying to open is not open"<<endl;  }  else {  myFile<<"TASK 3 IS :"<<endl;  for (i = 1; i <= rows; i++) {  if (i % 2 == 1) {  for (j = 1; j <= rows - i; j++) {  cout << " ";  myFile<< " ";  }  d = num1;   for (j = 1; j <= i; j++) {  cout << num1;  myFile<<num1;  }  num1 += 2;  }  if (i % 2 == 0) {  for (j = 1; j <= rows - i; j++) {  cout << " ";  myFile<< " ";  }  d = i;  for (j = 1; j <= i; j++) {  cout << "#";  myFile<< "#";  }  }   cout << endl;  myFile<< endl;  }  }  myFile<<endl;  myFile.close(); } void printZero(int &num,fstream &myFile) {  int j;  int d;  myFile.open("shapesTask1.txt",ios::app);   if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile<< "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 4 IS :"<<endl;  for (int i = num; i > 0; i--) {  for (j = 1; j <= i; j++) {  cout << "1";  myFile<< "1";  }   d = num;  for (j = 1; j <= num - i; j++) {  cout << "0";  myFile<< "0";  }   cout << endl;  myFile <<endl;  }  }  myFile<<endl;  myFile.close(); }  void printZero1(int &rows,fstream &myFile,ifstream &files) {  int j;  int d;  myFile.open("shapesTask1.txt",ios::app);   if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile<< "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 4 IS :"<<endl;  for (int i = rows; i > 0; i--) {  for (j = 1; j <= i; j++) {  cout << "1";  myFile<< "1";  }   d = rows;  for (j = 1; j <= rows - i; j++) {  cout << "0";  myFile<< "0";  }   cout << endl;  myFile <<endl;  }  }  myFile<<endl;  myFile.close(); }  void printChars(int &num,fstream &myFile) {  int j;  int d;  myFile.open("shapesTask1.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile<< "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 5 IS :"<<endl;  for (int i = num; i > 0; i--) {  for (j = 0; j <= num - i; j++) {  cout << "#";  myFile<< "#";  }  d = num;   for (j = 1; j <= i; j++) {  cout << "$";  myFile<< "$";  }  cout << endl;  myFile<<endl;  }  }  myFile<<endl;  myFile.close(); } void printChars1(int &rows,fstream &myFile,ifstream &files) {  int j;  int d;  myFile.open("shapesTask1.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile<< "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 5 IS :"<<endl;  for (int i = rows; i > 0; i--) {  for (j = 0; j <= rows - i; j++) {  cout << "#";  myFile<< "#";  }  d = rows;   for (j = 1; j <= i; j++) {  cout << "$";  myFile<< "$";  }  cout << endl;  myFile<<endl;  }  }  myFile<<endl;  myFile.close(); } void printCharOfTwoTypes(int &num,fstream &myFile) {  int j;  int d;  myFile.open("shapesTask1.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 6 IS :"<<endl;  for (int i = 1; i <= num; i++) {  for (int j = 1; j <=i; j++) {  cout << "#";  myFile<< "#";  }  cout << "$";  myFile<< "$";   for (int k = i; k <=num; k++) {  cout << "#";  myFile<< "#";  }  cout << endl;  myFile <<endl;   }  }  myFile<<endl;  myFile.close(); } void printCharOfTwoTypes1(int &rows,fstream &myFile,ifstream &files) {  int j;  int d;  myFile.open("shapesTask1.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 6 IS :"<<endl;  for (int i = 1; i <= rows; i++) {  for (int j = 1; j <=i; j++) {  cout << "#";  myFile<< "#";  }  cout << "$";  myFile<< "$";   for (int k = i; k <rows; k++) {  cout << "#";  myFile<< "#";  }  cout << endl;  myFile <<endl;   }  }  myFile<<endl;  myFile.close(); } |

**Task B**

Write a C++ code to output different shapes using the concept of nested

loops and file handling in multiple steps given below:

* Take input from a file
* Take input from a user
* Pass this information to a function to print shapes
* Write these shapes to a file

Chart, scatter chart

Description automatically generated

|  |
| --- |
| void rightAngle(int &size,fstream &myFile) {  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 7 IS :"<<endl;  for (int i = 1; i <= size; i++) {  for (int j = 1; j <= i; j++) {  cout << j;  myFile << j;  }  cout << endl;  myFile << endl;  }  }  myFile<<endl;  myFile.close(); } void rightAngle1(int &rows,fstream &myFile,ifstream &files) {  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 7 IS :"<<endl;  for (int i = 1; i <= rows; i++) {  for (int j = 1; j <= i; j++) {  cout << j;  myFile << j;  }  cout << endl;  myFile << endl;  }  }  myFile<<endl;  myFile.close(); } void invertedRightTriangle(int &size,fstream &myFile) {  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 8 IS :"<<endl;  for (int j = size; j >= 1; j--) {  for (int i = 1; i <= j; i++) {  cout << i;  myFile << i;  }  cout << endl;  myFile << endl;  }  }  myFile<<endl;  myFile.close(); } void invertedRightTriangle1(int &rows,fstream &myFile,ifstream &files) {  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 8 IS :"<<endl;  for (int j = rows; j >= 1; j--) {  for (int i = 1; i <= j; i++) {  cout << i;  myFile << i;  }  cout << endl;  myFile << endl;  }  }  myFile<<endl;  myFile.close(); } void rightHollowTriangle(int &size,fstream &myFile) {  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 9 IS :"<<endl;  for (int i = 1; i <= size; i++) {  for (int j = 1; j <= i; j++) {  if (i == size) {  cout << j;  myFile << j;  } else {  if (j == 1 || j == i) {  cout << j;  myFile << j;  } else {  cout << " ";  myFile<< " ";  }  }  }  cout << endl;  myFile << endl;  }  }  myFile<<endl;  myFile.close(); } void rightHollowTriangle1(int &rows,fstream &myFile,ifstream &files) {  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 9 IS :"<<endl;  for (int i = 1; i <= rows; i++) {  for (int j = 1; j <= i; j++) {  if (i == rows) {  cout << j;  myFile << j;  } else {  if (j == 1 || j == i) {  cout << j;  myFile << j;  } else {  cout << " ";  myFile<< " ";  }  }  }  cout << endl;  myFile << endl;  }  }  myFile<<endl;  myFile.close(); } void CompleteTriangle(int &size,fstream &myFile) {  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 10 IS :"<<endl;  int num = 0;  int num1 = 0;  int k = 0;  int i = 0;  while (i < size) {  i++;  for (int j = 1; j <= size - i; j++) {  cout << " ";  myFile<< " ";  num++;  }  for (k = 0; k != 2 \* i - 1; k++) {  if (num <= size - 1) {  cout << (i + k) << " ";  myFile << (i + k) << " ";  num++;  } else {  num1++;  cout << ((i + k) - (2 \* num1)) << " ";  myFile<<((i + k) - (2 \* num1)) << " ";  }  }  cout << endl;  myFile << endl;  num1 = num = k = 0;  }  }  myFile<<endl;  myFile.close(); } void CompleteTriangle1(int &rows,fstream &myFile,ifstream &files) {  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 10 IS :"<<endl;  int num = 0;  int num1 = 0;  int k = 0;  int i = 0;  while (i < rows) {  i++;  for (int j = 1; j <= rows - i; j++) {  cout << " ";  myFile<< " ";  num++;  }  for (k = 0; k != 2 \* i - 1; k++) {  if (num <= rows - 1) {  cout << (i + k) << " ";  myFile << (i + k) << " ";  num++;  } else {  num1++;  cout << ((i + k) - (2 \* num1)) << " ";  myFile<<((i + k) - (2 \* num1)) << " ";  }  }  cout << endl;  myFile << endl;  num1 = num = k = 0;  }  }  myFile<<endl;  myFile.close(); }   void hollowCompleteTriangle(int &size,fstream &myFile) {  int num = size;  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 11 IS :"<<endl;  for (int i = 0; i <= size; i++) {  for (int j = 0; j < num; j++) {  cout << " ";  myFile<< " ";  }  num--;  for (int k = 1; k <= i; k++) {  if (k == 1 || k == i || k == size || i == size) {  cout << k << " ";  myFile<<k<< " ";  } else {  cout << " ";  myFile<< " ";  }  }  cout << endl;  myFile << endl;  }  }  myFile<<endl;  myFile.close(); } void hollowCompleteTriangle1(int &rows,fstream &myFile,ifstream &files) {  int num = rows;  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 11 IS :"<<endl;  for (int i = 0; i <= rows; i++) {  for (int j = 0; j <rows; j++) {  cout << " ";  myFile<< " ";  }  num--;  for (int k = 1; k <= i; k++) {  if (k == 1 || k == i || k == rows || i == rows) {  cout << k << " ";  myFile<<k<< " ";  } else {  cout << " ";  myFile<< " ";  }  }  cout << endl;  myFile << endl;  }  }  myFile<<endl;  myFile.close(); } void invertedRightTriangleHollow(int &size,fstream &myFile) {  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 12 IS :"<<endl;  for (int i = size; i >= 1; i--) {  for (int j = 1; j <= i; j++) {  cout << size - i + j;  myFile << size - i + j;  }  cout << endl;  myFile << endl;  }  }  myFile<<endl;  myFile.close(); } void invertedRightTriangleHollow1(int &rows,fstream &myFile,ifstream &files) {  myFile.open("shapesTask2.txt",ios::app);  if (!myFile.is\_open()) {  cout << "the file you are trying to open is not open" << endl;  myFile << "the file you are trying to open is not open" << endl;  } else {  myFile<<"TASK 12 IS :"<<endl;  for (int i = rows; i >= 1; i--) {  for (int j = 1; j <= i; j++) {  cout << rows - i + j;  myFile << rows - i + j;  }  cout << endl;  myFile << endl;  }  }  myFile<<endl;  myFile.close(); } |

#### **Assessment Rubric for Lab**

**Method for assessment:**

Lab reports and instructor observation during lab sessions. Outcome assessed:

a. Ability to conduct experiments, as well as to analyze and interpret data (P) b. Ability to function on multi-disciplinary teams (A)

c. Ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (P)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Performance metric** | **Task** | **CLO** | **Description** | **Max marks** | **Exceeds expectation** | **Meets expectation** | **Does not meet expectation** | **Obtained marks** |
| 1. Realization of experiment (a) | 1 | 1 | Functionality | 40 | Executes without errors excellent user prompts, good use of symbols, spacing in output. Through testing has been completed (35-40) | Executes without errors, user prompts are understandable, minimum use of symbols or spacing in output. Some testing has been completed (20-34) | Does not execute due to syntax errors, runtime errors, user prompts are misleading or non-existent. No testing has been completed (0-19) |  |
| 2. Teamwork (b) | 1 | 3 | Group Performance | 5 | Actively engages and cooperates with other group member(s) in effective manner (4-5) | Cooperates with other group member(s) in a reasonable manner but conduct can be improved (2-3) | Distracts or discourages other group members from conducting the experiment (0-1) |  |
| 3. Conducting experiment (a, c) | 1 | 1 | On Spot Changes | 10 | Able to make changes (8-10) | Partially able to make changes (5-7) | Unable to make changes (0-4) |  |
| 1 | 1 | Viva | 10 | Answered all questions (8-10) | Few incorrect answers (5-7) | Unable to answer all questions (0-4) |  |
| 4. Laboratory safety and disciplinary rules (a) | 1 | 3 | Code commenting | 5 | Comments are added and does help the reader to understand the code (4-5) | Comments are added and does not help the reader to understand the code (2-3) | Comments are not added (0-1) |  |
| 5. Data collection (c) | 1 | 3 | Code Structure | 5 | Excellent use of white space, creatively organized work, excellent use of variables and constants, correct identifiers for constants, No line-wrap (4-5) | Includes name, and assignment, white space makes the program fairly easy to read. Title, organized work, good use of variables (2-3) | Poor use of white space (indentation, blank lines) making code hard to read, disorganized and messy (0-1) |  |
| 6. Data analysis (a, c) | 1 | 4 | Algorithm | 20 | Solution is efficient, easy to understand, and maintain (15-20) | A logical solution that is easy to follow but it is not the most efficient (6-14) | A difficult and inefficient solution (0-5) |  |
| 7. Computer use (c) | 1 | 2 | Documentation & Github Submissions | 5 | Timely (4-5) | Late (2-3) | Not done (0-1) |  |
|  | Max Marks (total): | | | 100 | Obtained Marks (total): | | |  |

Lab Engineer Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_