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

|  |  |
| --- | --- |
| **Course Instructor: Usama Bin Shakeel** | **Dated: 15/03/2022** |
| **Teaching Assistant: Aqsa Khalid** | **Semester: Spring 2022** |
| **Lab Engineer: Nadir Abbas** | **Batch: BSCE2021** |

# **Lab 2A. Output Different Shapes using Nested Loops**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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.

## **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.

**Lab Task**

**Task A**

Write a C++ program to generate the following patterns using nested loops

Take input from a user

Pass this information to a function to print patterns

Write a function for each pattern and call the function in switch statement

Calendar

Description automatically generated

|  |
| --- |
| void oddRightTriangle() {  int num;  cout << "please enter number of rows =" << endl;  cin >> num;  int j=0; for(int i=1;i<=num;i++){  if(i%2==1){  for(j=0;j<i;j++) cout<<i;  }  if(i%2==0){  for(int k=0;k<=j;k++) cout<<"#";  }  cout<<endl; } } void invertedRight(){  int num1;  cout<<"enter num1= "<<endl;  cin>>num1;  for(int j=1;j<=num1;j++){  for(int i=num1;i>=j;i--){  if(j%3==1){  cout<<"#";  }  if(j%3==2){  cout<<i;  }  if(j%3==0){  cout<<"$";  }  }  cout<<endl;  } } void evenRightTriangle() {  int num;  cout << "please enter number of rows =" << endl;  cin >> num;  int d;  int i;  int j;  int num1=2;  for (i = 1; i <= num; i++)  {  if(i%2==1){  for (j = 1; j <= num - i; j++)  {  cout <<" ";  }  d = num1;   for (j = 1; j <= i; j++)  {  cout<<num1;  }  num1+=2;  }  if(i%2==0){  for (j = 1; j <= num - i; j++)  {  cout <<" ";  }  d = i;  for (j = 1; j <= i; j++)  {  cout <<"#";  }  }   cout << endl;  } } void printZero(){  int n;  int j;  int d;  cout<<"please enter n";  cin>>n;  for (int i = n; i > 0; i--) {  for (j = 1; j <=i; j++)  {  cout<<"1";  }   d = n;  for (j = 1; j <= n - i; j++)  {  cout <<"0";  }   cout<<endl;  } }  void printChars(){  int n;  int j;  int d;  cout<<"please enter n";  cin>>n;  for (int i = n; i > 0; i--) {  for (j = 0; j <= n - i; j++)  {  cout <<"#";  }  d = n;   for (j = 1; j <= i; j++)  {  cout<<"$";  }  cout<<endl;  } }  void printCharOfTwoTypes() {  int n;  int j;  int d;  cout << "please enter num =";  cin >> n;  for (int i = 1; i <= n; i++) {  for (int j = 1; j < i; j++) {  cout << "#";  }  cout << "$";   for (int k = i; k < n; k++) {  cout << "#";  }  cout << endl;   } }  MAIN.CPP:  int opt; int opt1; cout<<"TASK 1"<<endl; cout<<"1.ODD RIGHT ANGLE "<<endl; cout<<"2.INVERTED "<<endl; cout<<"3.EVEN RIGHT ANGLE"<<endl; cout<<"4.PRINT ZERO"<<endl; cout<<"5.PRINT CHAR"<<endl; cout<<"6.PRINT CHAR OF TWO TYPES"<<endl; cout<<"7.EXIT"<<endl; cin>>opt; switch(opt){  case 1:{  oddRightTriangle();  break;  }  case 2:{  invertedRight();  break;  }  case 3:{  evenRightTriangle();  break;   }  case 4:{  printZero();  break;   }  case 5:{  printChars();  break;   }  case 6:{  printCharOfTwoTypes();  break;   }  case 7:{  cout<<"you choose to exit"<<endl;  break;   }  default:{  cout<<"you have entered invalid input"<<endl;  break; |

**Task B**

Write a C++ program to generate the following patterns using nested loops

Take input from a user

Pass this information to a function to print patterns

Write a function for each pattern and call the function in switch statement

Chart, scatter chart

Description automatically generated

|  |
| --- |
| void rightAngle(int &size){  for(int i=1;i<=size;i++){  for(int j=1;j<=i;j++){  cout<<j;  }  cout<<endl;  } } void invertedRightTriangle(int &size){  for(int j=size;j>=1;j--){  for(int i=1;i<=j;i++){  cout<<i;  }  cout<<endl;  } } void rightHollowTriangle(int &size) {  for (int i = 1; i <= size; i++) {  for (int j = 1; j <= i; j++) {  if (i == size) {  cout <<j;  }  else{  if (j == 1 || j == i) {  cout <<j;  } else {  cout<<" ";  }  }  }  cout<<endl;  }  } void CompleteTriangle(int &size) {  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 << " ";  num++;  }  for(k=0;k!=2\*i-1;k++) {  if (num <= size - 1) {  cout << (i + k )<< " ";  num++;  } else {  num1++;  cout << ((i + k )- (2 \* num1)) << " ";  }  }  cout << endl;  num1 = num = k = 0;  } }  void hollowCompleteTriangle(int &size){ int num=size; for(int i=0;i<=size;i++){  for(int j=0;j<num;j++){  cout<<" ";  }  num--;  for(int k=1;k<=i;k++){  if(k==1 || k==i || k==size || i==size){  cout<<k<<" ";  }  else{  cout<<" ";  }  }  cout<<endl; }  } void invertedRightTriangleHollow(int &size) { for(int i=size;i>=1;i--){  for(int j=1;j<=i;j++){  cout<<size-i+j;  }  cout<<endl; } }  MAIN.CPP:  cout<<"TASK 2"<<endl;  cout<<"1. RIGHT ANGLE "<<endl;  cout<<"2.INVERTED RIGHT ANGLE "<<endl;  cout<<"3.HOLLOW RIGHT ANGLE"<<endl;  cout<<"4.COMPLETE TRIANGLE"<<endl;  cout<<"5.HOLLOW COMPLETE"<<endl;  cout<<"6.INVERTED"<<endl;  cout<<"7.EXIT"<<endl;  cin>>opt1;  int size;  cout<<"please enter size ="<<endl;  cin>>size;  switch(opt1){  case 1:{  rightAngle(size);  break;  }  case 2:{  invertedRightTriangle(size);  break;  }  case 3:{  rightHollowTriangle( size);  break;   }  case 4:{  CompleteTriangle( size);  break;   }  case 5:{  hollowCompleteTriangle(size);  break;   }  case 6:{  invertedRightTriangleHollow( size);  break;   }  case 7:{  cout<<"you choose to exit"<<endl;  break;   }  default:{  cout<<"you have entered invalid input"<<endl;  break;   }  }  return 0; } |

#### **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: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_