| ***Computer Engineering Department*** |
| --- |
| ***CE100L: Computing Fundamentals & Programming*** |

| ***Course Instructor: Usama Bin Shakeel*** | ***Dated: 12/11/2021*** |
| --- | --- |
| ***Teaching Assistant: Aqsa Khalid*** | ***Semester: Fall 2021*** |
| ***Lab Engineer: Nadir Abbas*** | ***Batch: BSCE2021*** |

# **Lab 7B. 2D Arrays**

| **Name** | **Roll number** | **Report**  **(out of 100)** | **Scaled to 10** | **Total**  **(out of 10)** |
| --- | --- | --- | --- | --- |
| Syed Afraz | BSCE21024 |  |  |  |

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

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

## **Objective**

The objective of this session is to learn the working and advantages of arrays in C++.

## **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 multi-dimensional array is an array of more than one array. For example, an array of two one-dimensional arrays is called a two dimensional array. It is also known as table or matrix.

A two dimensional array consists of columns and rows. Each element of the two dimensional array is referenced by its index values or subscripts.

A two dimensional array is declared by giving two indexed values enclosed in square brackets. The first indexed value represents the total number of rows and the second represents the total number of columns. The syntax to declare a two dimensional array is: **type array\_name [r][c];**

where type represents the data type of the array e.g. int, float, double, char etc. array\_name represents the name of a two dimensional array. r represents the total number of rows of table. It is an unsigned number. c represents the total number of columns of the table. It is an unsigned number.

**Lab Task**

Write a program in C++ to input values into a table. Find out the total number of odd and even values entered in the array. Print these on the screen.

### Write pseudocode

| Int array[rows][columns],  Set constant rows=10  Set constant columns=10  For(int i=0;i<rows;i++)  {  For(int j=0;j<columns;j++)  {  Print enter the values of I and j  Read the values of I and j  }  }  for(int e=0;e<rows;e++)  {  for(int f=0;f<cols;f++)  {  cout<<ar[e][f]<<"\t";  }  cout<<endl;  }  int odd=0;  int even=0;  for(int g=0;g<rows;g++)  {  for(int h=0;h<cols;h++)  {  if(ar[g][h]%2==0)  {  even=even+1;  }else  odd=odd+1;  }  }  cout<<"The number of even digits are:"<<even;  cout<<endl;  cout<<"The number of odd digits are:"<<odd;  int ar[rows][cols];  for(int o=0;o<rows;o++)  {  for(int p=0;p<cols;p++)  {  cout<<"Insert the value of"<<"["<<o<<"]"<<"["<<p<<"]";  cin>>ar[o][p];  cout<<endl;  }  }  cout<<"Here are the values arranged in a table form"<<endl;  tab(ar);  tot(ar); |
| --- |

1. Draw flowchart

|  |
| --- |

1. Write C++ program

| void table(int arr[rows][columns])  {  for(int x=0;x<rows;x++) //applying nested loop of for to display the array  {  for(int y=0;y<columns;y++)  {  cout<<arr[x][y]<<"\t"; //displaying array  }  cout<<endl;  }  }  void totals(int arr[rows][columns])  {  int totalOdd=0; //initializing totalOdd  int totalEven=0; //initializing totalEven  for(int k=0;k<rows;k++)  {  for(int l=0;l<columns;l++) //applying nested loop  {  if(arr[k][l]%2==0 ) // applying condition that if the mode of array is equal to zero  {  totalEven=totaleven+1; //adding 1 as array start from zero  }else  totalodd=totalOdd+1; //if mode is not equal to zero them add 1 in total odd  }  }  cout<<"The Total number of even digits are:"<<totalEven; //displaying even digits  cout<<endl;  cout<<"The Total number of odd digits are:"<<totalOdd; //displaying odd digits  } |
| --- |

|  |
| --- |

#### **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 | Mapping (task no. and description) | | Max marks | Exceeds expectation | Meets expectation | Does not meet expectation | Obtained marks |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Realization of experiment (a) | 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 | 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 | On Spot Changes | 10 | Able to make changes (8-10) | Partially able to make changes (5-7) | Unable to make changes (0-4) |  |
| 2 | 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 | Code commenting | 5 | Observes lab safety rules; handles the equipment and parts with care and adheres to the lab disciplinary guidelines aptly (4-5) | Generally observes safety rules and disciplinary guidelines with minor lapses (2-3) | Disregards lab safety and disciplinary rules (0-1) |  |
| 5. Data collection (c) | 1 | 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 | 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 | Documentation | 5 | Timely documented (4-5) | Late documented (2-3) | Not documented (0-1) |  |
|  | Max Marks (total): | | 100 | Obtained Marks (total): | | |  |

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