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

| ***Course Instructor: Usama Bin Shakeel*** | ***Dated: 29/10/2021*** |
| --- | --- |
| ***Teaching Assistant: Aqsa Khalid*** | ***Semester: Fall 2021*** |
| ***Lab Engineer: Nadir Abbas*** | ***Batch: BSCE2021*** |

# **Lab 5B. Switch Cases in C++**

| **Name** | **Roll number** | **Report**  **(out of 100)** | **Scaled to 10** | **Total**  **(out of 10)** |
| --- | --- | --- | --- | --- |
| Nimra Maqbool | Bsce21012 |  |  |  |

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

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

## **Objective**

The goal of this handout is to learn the working of loops 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 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.

Switch case statement is used when we have multiple conditions and we need to perform different action based on the condition. When we have multiple conditions and we need to execute a block of statements when a particular condition is satisfied. In such case either we can use lengthy [if..else-if statement](https://beginnersbook.com/2017/08/cpp-if-else-statement/) or switch case.

**Lab Tasks**

Three shapes Rectangle, Triangle and Circle are provided to user. You should ask the user to pick a shape out of three. Then give option to choose one from the following:

* Area
* Volume
* Perimeter

Your program should be menu driven, use switch cases to achieve the above mentioned functionality. Take required inputs from user according to the chosen operation.

**Task A**

Write an algorithm for above given problem

| This program is menu driven and uses switch cases and takes the required input for the chosen program.  initialize input, ch area, perimeter,volume,radius(r),length(l),width(w),base(b),l2  Show the menu in which we ask about the shape(circle or rectangle or triangle)  Ask the user about his choice  Then use 3 switch cases  1 switch case for areas of all three shapes  2 switch case for volume of all three shapes  3 switch case for perimeter of all three shapes  Then ask the user input according to the chosen program and the display output. |
| --- |

**Task B**

Draw a flowchart of Task A

|  |
| --- |

**Task C**

Write a C++ program of Task A

| float menu() {  int input; // initializing input  char ch; //initializing ch  double area, perimeter, volume, r, l, w, h, b, l2; //initializing double  cout << "menu" << endl; //displaying menu to ask user its choice's input  cout << "enter 1 for circle" << endl;  cout << "enter 2 for rectangle" << endl;  cout << "enter 3 for triangle" << endl;  cout << "enter your choice:";  cin >> input; //taking input  cout << input; //displaying input  switch (input) {  case 1: //if user enter 1 in choice then it will come in switch 1  cout << "what do you want to find" << endl;  cout << "enter a for area" << endl;  cout << "enter v for volume" << endl;  cout << "enter p for perimeter" << endl;  cout << "enter your choice:";  cin >> ch; //entering choice  cout << ch << endl; //displaying choice  switch (ch) {  case 'a':  cout << "enter radius of circle" << endl;  cin >> r;  area = 3.14 \* r \* r;  cout << "area of circle is " << area << endl; //displaying radius  break;  case 'v':  cout << "no volume of circle" << endl;  volume = 0;  cout << "volume of circle " << volume; //displaying volume  break;  case 'p':  cout << "enter radius of circle" << endl;  cin >> r;  perimeter = 2 \* 3.14 \* r;  cout << "perimeter of circle " << perimeter << endl; //displaying perimeter  }  break;  case 2: //if user enter 2 in menu choice then it will come in case 2  cout << "what do you want to find" << endl;  cout << "enter a for area" << endl;  cout << "enter v for volume" << endl;  cout << "enter p for perimeter" << endl;  cout << "enter your choice" << endl;  cin >> ch;  switch (ch) {  case 'a':  cout << "enter length of rectangle" << endl;  cin >> l;  cout << "enter width of rectangle" << endl;  cin >> w;  area = l \* w;  cout << "area of rectangle " << area << endl; //displaying area  break;  case 'v':  cout << "enter length" << endl;  cin >> l;  cout << "enter width " << endl;  cin >> w;  cout << "enter height" << endl;  cin >> h;  volume = l \* w \* h;  cout << "volume of rectangle " << volume << endl; //displaying volume  break;  case 'p':  cout << "enter length" << endl;  cin >> l;  cout << "enter width" << endl;  cin >> w;  perimeter = (l + w) \* 2;  cout << "perimeter of rectangle " << perimeter << endl; //displaying perimeter  }  break;  case 3: //if user enter 3 in choice then user will come in case 3  cout << "what do you want to find" << endl;  cout << "enter a for area" << endl;  cout << "enter v for volume" << endl;  cout << "enter p for perimeter" << endl;  cout << "enter your choice" << endl;  cin >> ch;  switch (ch) {  case 'a':  cout << "enter base" << endl;  cin >> l;  cout << "enter height" << endl;  cin >> h;  area = b \* h / 2;  cout << "area of triangle" << area << endl; //displaying area  break;  case 'v':  cout << "enter length" << endl;  cin >> l;  cout << "enter base" << endl;  cin >> b;  cout << "enter height" << endl;  cin >> h;  volume = l \* b \* h;  cout << "volume of rectangle " << volume << endl; //displaying volume  break;  case 'p':  cout << "enter base" << endl;  cin >> b;  cout << "enter length of one side" << endl;  cin >> l;  cout << "enter length of other side" << endl;  cin >> l2;  perimeter = b + l + l2;  cout << "perimeter of triangle " << perimeter << endl; //displaying perimeter  }  }  return 1.2;  } |
| --- |

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