ASSIGNMENT

TUTORIAL 01

Q1) simple mini calculator program in C++ that uses subroutines for basic arithmetic operations

CODE:

```
#include <iostream>
using namespace std;
double add(double a, double b);
double subtract(double a, double b);
double multiply(double a, double b);
double divide(double a, double b);

int main() {
    double n1, n2;
    char op;

    cout << "Enter first number: ";
    cin >> n1;

    cout << "Enter an operation (+, -, *, /): ";
    cin >> op;

    cout << "Enter second number: ";
    cin >> n2;
```

```
    double result;

    switch (op) {
        case '+':
            result = add(n1, n2);
            break;
        case '-':
            result = subtract(n1, n2);
            break;
        case '*':
            result = multiply(n1, n2);
            break;
        case '/':
            if (n2 != 0) {
                result = divide(n1, n2);
            } else {
                std::cout << "Error: Cannot divide by zero." << std::endl;
                return 1;
            }
            break;
        default:
            cout << "Invalid operation." << endl;
            return 1;
    }
    cout << "Result: " << result << endl;
    return 0;
}

double add(double a, double b) {
```

```
    return a + b;
}

double subtract(double a, double b) {
    return a - b;
}

double multiply(double a, double b) {
    return a * b;
}

double divide(double a, double b) {
    return a / b;
}
```


Output

```
/tmp/A3HXPyCb79.o
Enter first number: 12
Enter an operation (+, -, *, /): +
Enter second number: 5
Result: 17
```

Q2) Write a complete students database with subroutines involves storing and managing student information using appropriate data structures and providing various functionalities to interact with the database and implement in C++ with subroutines:

CODE:

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

struct Student {
    string name;
    int rollNumber;
    double gpa;
};

void addStudent(vector<Student>& database);
void displayStudents(const vector<Student>& database);
void searchStudent(const vector<Student>& database, int rollNumber);

int main() {
    vector<Student> studentDatabase;

    while (true) {
        cout << "Menu:" << endl;
        cout << "1. Add Student" << endl;
        cout << "2. Display Students" <<endl;
        cout << "3. Search Student by Roll Number" <<endl;
```

```
        cout << "4. Exit" <<endl;

        int choice;
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                addStudent(studentDatabase);
                break;
            case 2:
                displayStudents(studentDatabase);
                break;
            case 3:
                int rollNumber;
                cout << "Enter roll number to search: ";
                cin >> rollNumber;
                searchStudent(studentDatabase, rollNumber);
                break;
            case 4:
                cout << "Exiting the program." <<endl;
                return 0;
            default:
                cout << "Invalid choice. Please enter a valid option." <<endl;
        }
    }

    return 0;
}
void addStudent(vector<Student>& database) {
    Student student;
    cout << "Enter student name: ";
```

```
    cin >> student.name;
    cout << "Enter student roll number: ";
    cin >> student.rollNumber;
    cout << "Enter student GPA: ";
    cin >> student.gpa;
    database.push_back(student);
    cout << "Student added successfully." << endl;
}


void displayStudents(const vector<Student>& database) {
    if (database.empty()) {
        cout << "No students in the database." <<endl;
        return;
    }

    cout << "List of students:" <<endl;
    for (const Student& student : database) {
        cout << "Name: " << student.name << ", Roll Number: " << student.rollNumber << ",
GPA: " << student.gpa <<endl;
    }
}

void searchStudent(const vector<Student>& database, int rollNumber) {
    for (const Student& student : database) {
        if (student.rollNumber == rollNumber) {
            cout << "Student found:" <<endl;
            cout << "Name: " << student.name << ", Roll Number: " << student.rollNumber << ",
GPA: " << student.gpa <<endl;
            return;
        }
    }
    cout << "Student with roll number " << rollNumber << " not found." <<endl;
```

}

```
Output
/tmp/ZztLoZM2oh.o
Menu:
1. Add Student
2. Display Students
3. Search Student by Roll Number
4. Exit
Enter your choice: 1
Enter student name: Gaurav
Enter student roll number: 284
Enter student GPA: 9.3
Student added successfully.
Menu:
1. Add Student
2. Display Students
3. Search Student by Roll Number
4. Exit
Enter your choice: 2
List of students:
Name: Gaurav, Roll Number: 284, GPA: 9.3
Menu:
1. Add Student
2. Display Students
3. Search Student by Roll Number
4. Exit
Enter your choice:
```

Q3) Design a subroutine program to calculate the area and perimeter of different geometric shapes (circle, rectangle, triangle, etc.).

CODE:

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

```
// Function prototypes
double calculateCircleArea(double radius);
double calculateCirclePerimeter(double radius);
double calculateRectangleArea(double length, double width);
double calculateRectanglePerimeter(double length, double width);
double calculateTriangleArea(double base, double height);
double calculateTrianglePerimeter(double side1, double side2, double side3);

int main() {
  int choice;

  do {
    cout << "Geometry Calculator" <<endl;
    cout << "1. Calculate Circle" <<endl;
    cout << "2. Calculate Rectangle" << endl;
    cout << "3. Calculate Triangle" << endl;
    cout << "4. Exit" << endl;

    cout << "Enter your choice: ";
    cin >> choice;

    switch (choice) {
      case 1:
        double radius;
        cout << "Enter the radius of the circle: ";
        cin >> radius;
        cout << "Area: " << calculateCircleArea(radius) << endl;
        cout << "Perimeter: " << calculateCirclePerimeter(radius) << endl;
        break;
      case 2:
        double length, width;
```

```
            cout << "Enter the length of the rectangle: ";
            cin >> length;
            cout << "Enter the width of the rectangle: ";
            cin >> width;
            cout << "Area: " << calculateRectangleArea(length, width) << endl;
            cout << "Perimeter: " << calculateRectanglePerimeter(length, width) << endl;
            break;
        case 3:
            double base, height;
            cout << "Enter the base of the triangle: ";
            cin >> base;
            cout << "Enter the height of the triangle: ";
            cin >> height;
            cout << "Area: " << calculateTriangleArea(base, height) << endl;
            break;
        case 4:
            cout << "Exiting the program." << endl;
            break;
        default:
            cout << "Invalid choice. Please enter a valid option." << endl;
    }
  } while (choice != 4);

  return 0;
}


double calculateCircleArea(double radius) {
  return M_PI * radius * radius;
}

double calculateCirclePerimeter(double radius) {
```

```
    return 2 * M_PI * radius;
}

double calculateRectangleArea(double length, double width) {
    return length * width;
}

double calculateRectanglePerimeter(double length, double width) {
    return 2 * (length + width);
}

double calculateTriangleArea(double base, double height) {
    return 0.5 * base * height;
}
double calculateTrianglePerimeter(double side1, double side2, double side3) {
    return side1 + side2 + side3;
}
```

Output

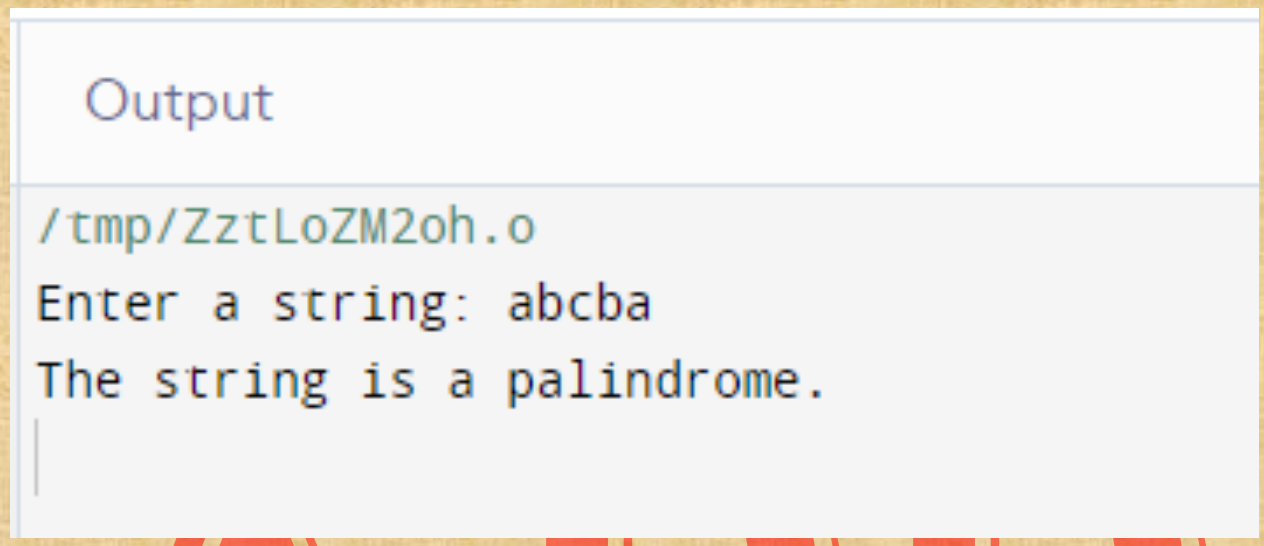
```
/tmp/ZztLoZM2oh.o
Geometry Calculator
1. Calculate Circle
2. Calculate Rectangle
3. Calculate Triangle
4. Exit
Enter your choice: 1
Enter the radius of the circle: 5
Area: 78.5398
Perimeter: 31.4159
Geometry Calculator
1. Calculate Circle
2. Calculate Rectangle
3. Calculate Triangle
4. Exit
Enter your choice:
```

Q4) Implement a subroutine program to check if a given string is a palindrome or not.

CODE:

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

// Function prototype
bool isPalindrome(const string& str);

int main() {
   string input;

   cout << "Enter a string: ";
   getline(cin, input);

   if (isPalindrome(input)) {
      cout << "The string is a palindrome." << endl;
   } else {
   cout << "The string is not a palindrome." <<endl;
   }

   return 0;
}

// Function to check if a string is a palindrome
bool isPalindrome(const string& str) {
   string reversed = str;
```

```
    reverse(reversed.begin(), reversed.end());

    return str == reversed;
}
```



Q5) Implement a subroutine program to reverse an array of integers in-place.

CODE

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

// Function prototype
void reverseArray(int arr[], int size);

int main() {
    int size;

    cout << "Enter the size of the array: ";
    cin >> size;
```

```
    int arr[size];

    cout << "Enter " << size << " integers:" << endl;
    for (int i = 0; i < size; ++i) {
        cin >> arr[i];
    }

    reverseArray(arr, size);

    cout << "Reversed array:" << endl;
    for (int i = 0; i < size; ++i) {
        cout << arr[i] << " ";
    }
    cout << std::endl;

    return 0;
}
void reverseArray(int arr[], int size) {
    int start = 0;
    int end = size - 1;

    while (start < end) {

        int temp = arr[start];
        arr[start] = arr[end];
        arr[end] = temp;


        ++start;
        --end;
    }
```

```
}
```

Output

```
/tmp/ZztLoZM2oh.o
Enter the size of the array: 2
Enter 2 integers:
1
2
Reversed array:
2 1
```