Domain Winter Winning Camp Day 1 Questions

Student Name: Nipun Chugh	**UID: 22BCS10636**
Branch: CSE	**Section/Group: KPIT-901/A**
Date of Performance:19/12/24	**Subject Name: C++/DSA**

Question 1: Calculate the sum of all natural numbers from 1 to n, where n is a positive integer. Use the formula:
Sum=n×(n+1)/2 .
Take n as input and output the sum of natural numbers from 1 to n .

Question 2: Count the total number of digits in a given number n. The number can be a positive integer. For example, for the number 12345, the count of digits is 5. For a number like 900000, the count of digits is 6. Given an integer n, your task is to determine how many digits are present in n. This task will help you practice working with loops, number manipulation, and conditional logic.

Question 3: Write a program to calculate the area of different shapes using function overloading. Implement overloaded functions to compute the area of a circle, a rectangle, and a triangle.

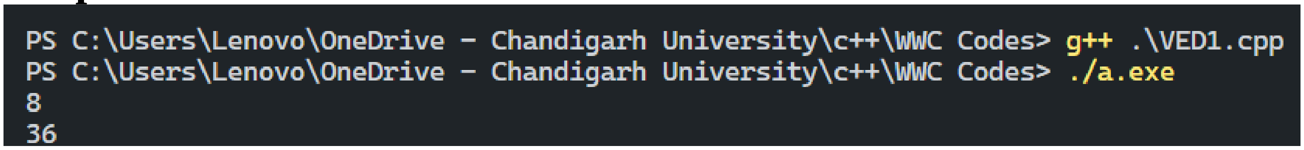
Question 4: Write a program to demonstrate runtime polymorphism in C++ using a base class Shape and derived classes Circle, Rectangle, and Triangle. The program should use virtual functions to calculate and print the area of each shape based on user input.

Question 5: Create a C++ program to simulate an employee management system using hierarchical inheritance. Design a base class Employee that stores basic details (name, ID, and salary). Create two derived classes: Manager: Add and calculate bonuses based on performance ratings. Developer: Add and calculate overtime compensation based on extra hours worked. The program should allow input for both types of employees and display their total earnings.

Solution 1:

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

        int main () {
            int n;
            cin>>n;
            int sum = n*(n+1)/2;
            cout<<sum<<endl;
            return 0;
}
```

Output:

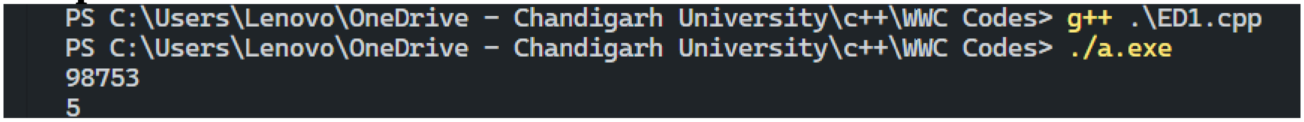


```
PS C:\Users\Lenovo\OneDrive - Chandigarh University\c++\WWC Codes> g++ .\VED1.cpp
PS C:\Users\Lenovo\OneDrive - Chandigarh University\c++\WWC Codes> ./a.exe
8
36
```

Solution 2:

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

int main () {
    int n;
    cin>>n;
    int DigitCount = 0;
    while(n>0){
        n /= 10; // N = n divided by 10
        DigitCount++;
    }
    cout<<DigitCount<<endl;
    return 0;
}
```

Output:



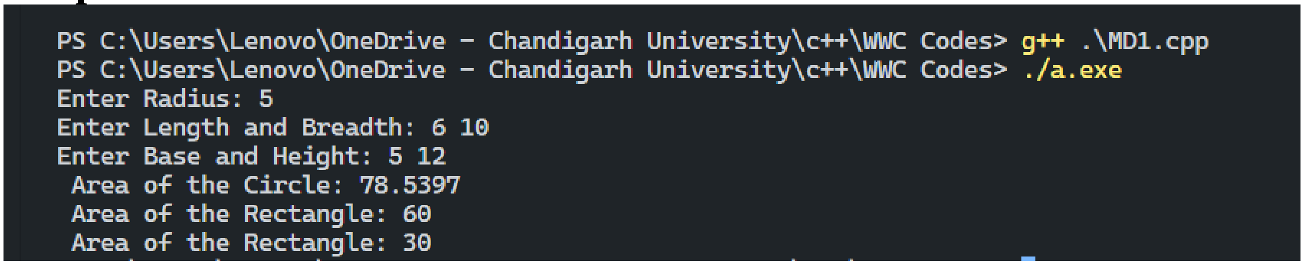
```
PS C:\Users\Lenovo\OneDrive - Chandigarh University\c++\WWC Codes> g++ .\ED1.cpp
PS C:\Users\Lenovo\OneDrive - Chandigarh University\c++\WWC Codes> ./a.exe
98753
5
```

Solution 3:

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

double CalculateArea(double radius){
    const double PI = 3.14159;
    return PI * radius * radius;
}

int CalculateArea(int length, int breadth){
    return length * breadth;
}
```

```
int CalculateArea(double base, double height){
    return 0.5 * base * height;
}

int main () {
    double radius;
    cout<<"Enter Radius: ";
    cin>>radius;

    int length , breadth;
    cout<<"Enter Length and Breadth: ";
    cin>>length>>breadth;

    double base, height;
    cout<<"Enter Base and Height: ";
    cin>>base>>height;

    cout<<" Area of the Circle: "<<CalculateArea(radius)<<endl;
    cout<<" Area of the Rectangle:
"<<CalculateArea(length,breadth)<<endl;
    cout<<" Area of the Rectangle: "<<CalculateArea(base,
height)<<endl;

    return 0;
}
```

Output:



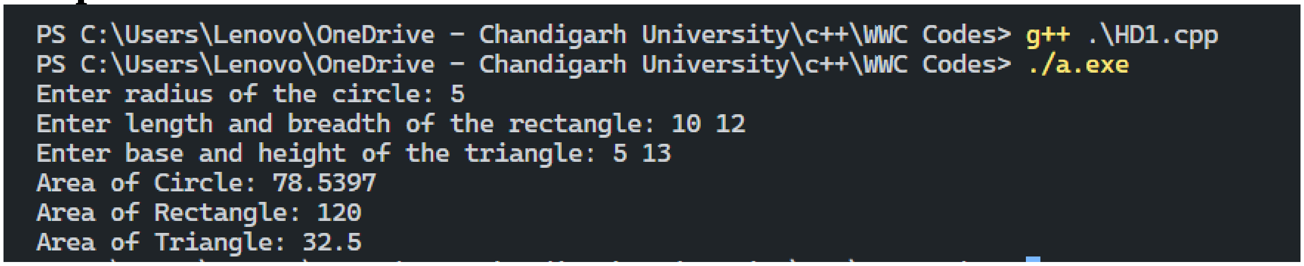
```
PS C:\Users\Lenovo\OneDrive - Chandigarh University\c++\WWC Codes> g++ .\MD1.cpp
PS C:\Users\Lenovo\OneDrive - Chandigarh University\c++\WWC Codes> ./a.exe
Enter Radius: 5
Enter Length and Breadth: 6 10
Enter Base and Height: 5 12
 Area of the Circle: 78.5397
 Area of the Rectangle: 60
 Area of the Rectangle: 30
```

Solution 4:

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

class Shape {
public:
    virtual void calculateArea() = 0;
    virtual ~Shape() {}
};

class Circle : public Shape {
private:
    double radius;
public:
    Circle(double r) : radius(r) {}
    void calculateArea() override {
        const double PI = 3.14159;
        double area = PI * radius * radius;
        cout << "Area of Circle: " << area << endl;
    }
};

class Rectangle : public Shape {
private:
    double length, breadth;
public:
    Rectangle(double l, double b) : length(l), breadth(b) {}
    void calculateArea() override {
        double area = length * breadth;
        cout << "Area of Rectangle: " << area << endl;
    }
};
```

```
class Triangle : public Shape {
private:
    double base, height;
public:
    Triangle(double b, double h) : base(b), height(h) {}
    void calculateArea() override {
        double area = 0.5 * base * height;
        cout << "Area of Triangle: " << area << endl;
    }
};

int main() {
    double radius;
    cout << "Enter radius of the circle: ";
    cin >> radius;
    Circle circle(radius);

    double length, breadth;
    cout << "Enter length and breadth of the rectangle: ";
    cin >> length >> breadth;
    Rectangle rectangle(length, breadth);

    double base, height;
    cout << "Enter base and height of the triangle: ";
    cin >> base >> height;
    Triangle triangle(base, height);

    Shape* shapes[3] = { &circle, &rectangle, &triangle };

    for (int i = 0; i < 3; ++i) {
        shapes[i]->calculateArea();
    }

    return 0;
}
```

Output:



```
PS C:\Users\Lenovo\OneDrive - Chandigarh University\c++\WWC Codes> g++ .\HD1.cpp
PS C:\Users\Lenovo\OneDrive - Chandigarh University\c++\WWC Codes> ./a.exe
Enter radius of the circle: 5
Enter length and breadth of the rectangle: 10 12
Enter base and height of the triangle: 5 13
Area of Circle: 78.5397
Area of Rectangle: 120
Area of Triangle: 32.5
```

Solution 5:

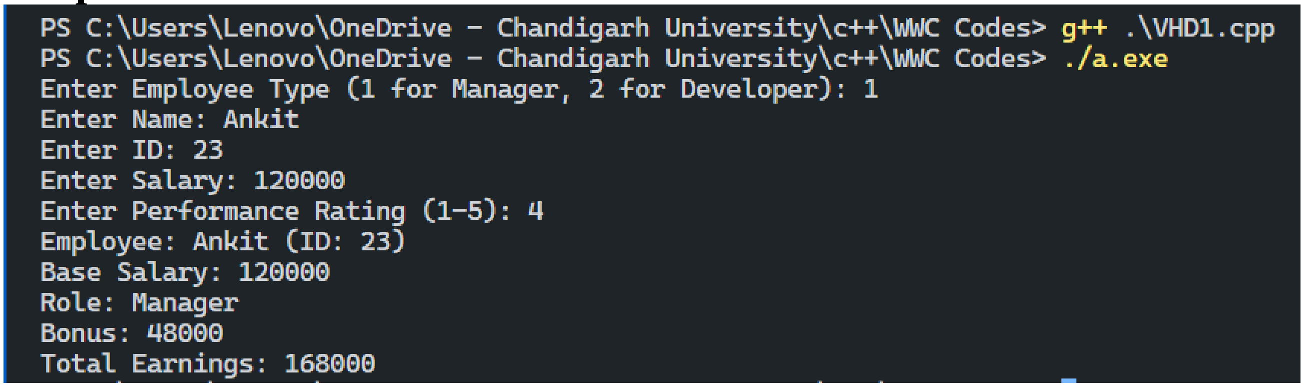
```
#include <iostream>
#include <string>
using namespace std;

// Base class: Employee
class Employee {
protected:
    string name;
    int id;
    int salary;

public:
    Employee(string empName, int empId, int empSalary)
        : name(empName), id(empId), salary(empSalary) {}

    virtual void calculateEarnings() = 0; // Pure virtual function

    virtual void displayDetails() const {
        cout << "Employee: " << name << " (ID: " << id << ")" <<
endl;
        cout << "Base Salary: " << salary << endl;
    }

    virtual ~Employee() {} // Virtual destructor
};
```

```
// Derived class: Manager
class Manager : public Employee {
private:
    int rating;

public:
    Manager(string empName, int empId, int empSalary, int
empRating)
        : Employee(empName, empId, empSalary), rating(empRating)
{}

    void calculateEarnings() override {
        double bonus = (rating * 0.1) * salary;
        displayDetails();
        cout << "Role: Manager" << endl;
        cout << "Bonus: " << bonus << endl;
        cout << "Total Earnings: " << salary + bonus << endl;
    }
};

// Derived class: Developer
class Developer : public Employee {
private:
    int extraHours;

public:
    Developer(string empName, int empId, int empSalary, int
empExtraHours)
        : Employee(empName, empId, empSalary),
extraHours(empExtraHours) {}

    void calculateEarnings() override {
        int overtimeCompensation = extraHours * 500;
        displayDetails();
        cout << "Role: Developer" << endl;
        cout << "Overtime Compensation: " << overtimeCompensation
<< endl;
```

```
        cout << "Total Earnings: " << salary + overtimeCompensation
<< endl;
    }
};

int main() {
    int empType;
    cout << "Enter Employee Type (1 for Manager, 2 for Developer):
";
    cin >> empType;

    if (empType == 1) {
        string name;
        int id, salary, rating;

        cout << "Enter Name: ";
        cin >> name;
        cout << "Enter ID: ";
        cin >> id;
        cout << "Enter Salary: ";
        cin >> salary;
        cout << "Enter Performance Rating (1-5): ";
        cin >> rating;

        if (rating < 1 || rating > 5) {
            cout << "Invalid rating. Please enter a value between 1 and
5." << endl;
            return 0;
        }

        Manager manager(name, id, salary, rating);
        manager.calculateEarnings();

    } else if (empType == 2) {
        string name;
        int id, salary, extraHours;
```

```
            cout << "Enter Name: ";
            cin >> name;
            cout << "Enter ID: ";
            cin >> id;
            cout << "Enter Salary: ";
            cin >> salary;
            cout << "Enter Extra Hours Worked: ";
            cin >> extraHours;

            if (extraHours < 0 || extraHours > 100) {
                cout << "Invalid hours. Please enter a value between 0 and
    100." << endl;
                return 0;
            }

            Developer developer(name, id, salary, extraHours);
            developer.calculateEarnings();

        } else {
            cout << "Invalid employee type." << endl;
        }

        return 0;
}
```

Output:



```
PS C:\Users\Lenovo\OneDrive - Chandigarh University\c++\WWC Codes> g++ .\VHD1.cpp
PS C:\Users\Lenovo\OneDrive - Chandigarh University\c++\WWC Codes> ./a.exe
Enter Employee Type (1 for Manager, 2 for Developer): 1
Enter Name: Ankit
Enter ID: 23
Enter Salary: 120000
Enter Performance Rating (1-5): 4
Employee: Ankit (ID: 23)
Base Salary: 120000
Role: Manager
Bonus: 48000
Total Earnings: 168000
```