



## DOMAIN WINTER WINNING CAMP

**Student Name:** Darshan  
**Branch:** BE-CSE  
**Semester:** 5th

**UID:** 22BCS15280  
**Section/Group:** 620-A  
**Date of Performance:** 19/12/24

### 1) Sum of Natural Numbers up to N

#### Objective:

Calculate the sum of all natural numbers from 1 to n, where n is a positive integer. Use the formula:

$$\text{Sum} = n \times (n+1) / 2$$

Take n as input and output the sum of natural numbers from 1 to n.

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

int main() {
    int n;
    cout<<"Darshan 22BCS15280 "<<endl;
    cout << "Enter n: ";    cin >> n;

    if (n > 0) {
        int sum = n * (n + 1) / 2;
        cout << "The sum of natural numbers from 1 to " << n << " is: " << sum << endl;
    }

    return 0;
}
```

```
Enter n: 100
The sum of natural numbers from 1 to 100 is: 5050

=== Code Execution Successful ===
```

### 2) Check if a Number is Prime

**Objective:**

Check if a given number  $n$  is a prime number. A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.

To determine if a number is prime, iterate from 2 to  $\sqrt{n}$  and check if  $n$  is divisible by any number in this range. If it is divisible, it is not a prime number; otherwise, it is a prime.

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

int main() {
    cout<<" Darshan 22BCS15280 "<<endl;    int n;
    cout << "enter a number: ";
    cin >> n;

    if (n <= 1) {
        cout << n << " is not a prime number." << endl;
    }
    else {
        for (int i = 2; i * i <= n; i++) {
            if (n % i == 0) {
                cout << n << " is not a prime number." << endl;
                return 0;
            }
        }
        cout << n << " is a prime number." << endl;
    }

    return 0;
}
```

```
enter a number: 73
73 is a prime number.

=== Code Execution Successful ===
```

**3) Print Odd Numbers up to N**

**Objective:**

**Print all odd numbers between 1 and n, inclusive. Odd numbers are integers that are not divisible by 2. These numbers should be printed in ascending order, separated by spaces. This problem is a simple introduction to loops and conditional checks. The goal is to use a loop to iterate over the numbers and check if they are odd using the condition  $i \% 2 \neq 0$ .**

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

int main() {
    cout<<" Darshan 22BCS15280 "<<endl;    int n;
    cout << "enter n: ";
    cin >> n;

    if (n > 0) {
        cout << "odd numbers up to " << n << " are:" << endl;
        for (int i = 1; i <= n; i += 2) {
            cout << i << " ";
        }
        cout << endl;
    }

    return 0;
}
```

```
enter n: 10
odd numbers up to 10 are:
1 3 5 7 9

=== Code Execution Successful ===
```

#### 4) Sum of Odd Numbers up to N

##### Objective:

**Calculate the sum of all odd numbers from 1 to n. An odd number is an integer that is not divisible by 2. The sum of odd numbers, iterate through all the numbers from 1 to n, check if each number is odd, and accumulate the sum.**

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

int main() {    cout<<" Darshan 22BCS15280
"<<endl;    int n;
    cout << "enter n: ";
    cin >> n;

    if (n > 0) {
int sum = 0;
        for (int i = 1; i <= n; i += 2) {
sum += i;
        }
        cout << "The sum of odd numbers up to " << n << " is: " << sum << endl;
    }

    return 0;
}
```

```
enter n: 10
The sum of odd numbers up to 10 is: 25

=== Code Execution Successful ===
```

## 5) Print Multiplication Table of a Number

### Objective:

**Print the multiplication table of a given number n. A multiplication table for a number n is a list of products of n with integers from 1 to 10. For example, the multiplication table for 3 is:  $3 \times 1 = 3, 3 \times 2 = 6, \dots, 3 \times 10 = 30$ .**

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

```
int main() {    cout<<" Darshan 22BCS15280
"<<endl;    int n;
    cout << "Enter a number: ";
    cin >> n;

    cout << "Multiplication table for " << n << " is:" << endl;
    for (int i = 1; i <= 10; ++i) {
        cout << n << " x " << i << " = " << n * i << endl;
    }

    return 0;
}
```

```
Enter a number: 11
Multiplication table for 11 is:
11 x 1 = 11
11 x 2 = 22
11 x 3 = 33
11 x 4 = 44
11 x 5 = 55
11 x 6 = 66
11 x 7 = 77
11 x 8 = 88
11 x 9 = 99
11 x 10 = 110

=== Code Execution Successful ===
```

## 6) Count Digits in a Number

### Objective:

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.

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



**DEPARTMENT OF**

**COMPUTER SCIENCE & ENGINEERING**

Discover. Learn. Empower.

```
int main() {
    int n;    int
    m = 0;
    cout<<" Darshan 22BCS15280 <<endl;
    cout << "enter n: ";    cin >> n;

    while (n > 0) {
        n = n / 10;
        m++;
    }

    cout << "no. of digits is " << m << endl;

    return 0;
}
```

```
enter n: 1234567890
no. of digits is 10
```

```
=== Code Execution Successful ===
```

## 7) Find the Largest Digit in a Number

### Objective:

**Find the largest digit in a given number n. For example, for the number 2734, the largest digit is 7. You need to extract each digit from the number and determine the largest one. The task will involve using loops and modulus operations to isolate the digits.**

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

```
int main() {
    cout<<" Darshan 22BCS15280 "<<endl;
    int n, m = 0;    cout<<"enter n: "<<endl;
    cin >> n;
```

```
    while (n > 0) {  
        int digit = n % 10;  
        if (digit > m) {  
            m = digit;  
        }  
        n = n / 10;  
    }  
  
    cout << m << endl;  
  
    return 0;  
}
```

```
enter n:  
6987  
9  
  
=== Code Execution Successful ===
```

## 8) Check if a Number is a Palindrome

### Objective

Check whether a given number is a palindrome or not. A number is called a palindrome if it reads the same backward as forward. For example, 121 is a palindrome because reading it from left to right is the same as reading it from right to left. Similarly, 12321 is also a palindrome, but 12345 is not.

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

```
int main() {  
    cout<<" Darshan 22BCS15280 "<<endl;  
    int n, original, reversed = 0;    cout<<"enter n:  
";    cin >> n;
```

```
original = n; while
(n > 0) {      int digit
= n % 10;
    reversed = reversed * 10 + digit;
    n = n / 10;
}

if (original == reversed) {
    cout << "The number is a palindrome." << endl;
} else {
    cout << "The number is not a palindrome." << endl;
}

return 0;
}
```

```
enter n: 141
The number is a palindrome.

=== Code Execution Successful ===
```

## 9) Find the Sum of Digits of a Number

### Objective:

**Calculate the sum of the digits of a given number n. For example, for the number 12345, the sum of the digits is  $1+2+3+4+5=15$ . To solve this, you will need to extract each digit from the number and calculate the total sum.**

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

int main() {

    cout<<" Darshan 22BCS15280 "<<endl;
```





```
int n, sum = 0;
cout<<"enter n: "; cin
>> n;

while (n > 0) {
int digit = n % 10;
sum += digit;      n /=
10;
}

cout << sum << endl;

return 0;
}
```

```
enter n: 1411
7
```

```
=== Code Execution Successful ===
```

## 10) Polymorphism with Shape Area Calculation.

### Objective:

Create a program that demonstrates polymorphism by calculating the area of different shapes using a base class Shape and derived classes for Circle, Rectangle, and Triangle. Each derived class should override a virtual function to compute the area of the respective shape.

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

```
class Shape { public:    virtual
double calculateArea() = 0;
};
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
class Circle : public Shape { private:
    double radius;

public:
    Circle(double r) : radius(r) {}
    double calculateArea() {
        return 3.14 * radius * radius;
    }
};

class Rectangle : public Shape { private:
    double length, width;

public:
    Rectangle(double l, double w) : length(l), width(w) {}
    double calculateArea() {
        return length * width;
    }
};

class Triangle : public Shape { private:
    double base, height;

public:
    Triangle(double b, double h) : base(b), height(h) {}
    double calculateArea() {
        return 0.5 * base * height;
    }
};

void displayArea(Shape* shape) {
    cout << "Area: " << shape->calculateArea() << endl;
}

int main() {
    cout<<" Darshan 22BCS15280 "<<endl;
    Circle c(5.0);
    Rectangle r(4.0, 6.0);
    Triangle t(3.0, 4.0);

    displayArea(&c);
    displayArea(&r);
    displayArea(&t);

    return 0;
}
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
}
```

```
Area: 78.5
```

```
Area: 24
```

```
Area: 6
```

```
=== Code Execution Successful ===|
```