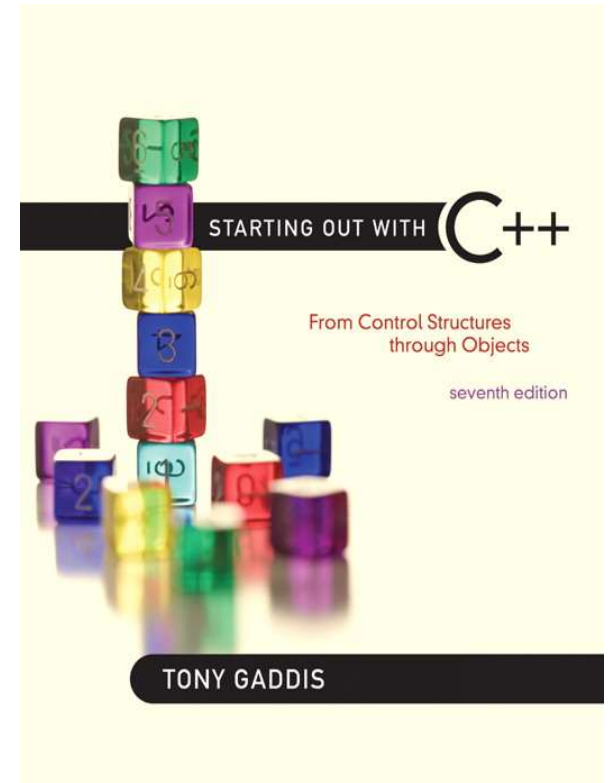


18 - 10 - 2022



# What will be the output

.....

```
int x = 2;  
cout<<x++ << --x;
```

.....

```
int c=10;  
int x = 2;  
cout<<x++ * --c<< x++ * --c;
```

# Some Examples using while loop

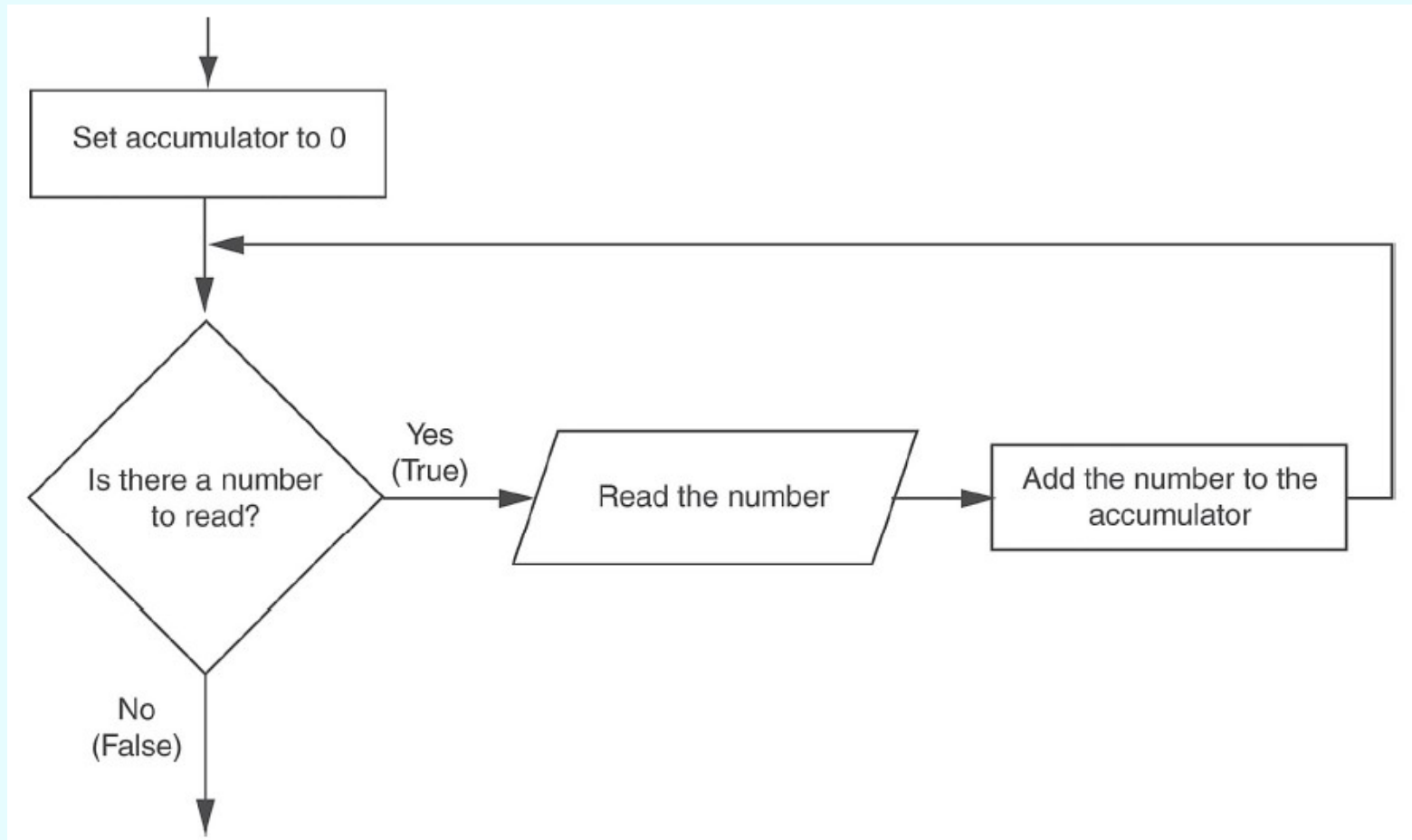
1. Printing different Series
2. Factorial
3. Table printing

# Keeping a Running Total

- running total: accumulated sum of numbers from each repetition of loop
- accumulator: variable that holds running total

```
int sum=0, num=1; // sum is the
while (num <= 10) // accumulator
{
    sum += num;
    num++;
}
cout << "Sum of numbers 1 – 10 is"
      << sum << endl;
```

# Logic for Keeping a Running Total



# A Running Total in Program 5-12

## Program 5-12

```
1  // This program takes daily sales figures over a period of time
2  // and calculates their total.
3  #include <iostream>
4  #include <iomanip>
5  using namespace std;
6
7  int main()
8  {
9      int days;           // Number of days
10     double total = 0.0; // Accumulator, initialized with 0
11
12     // Get the number of days.
13     cout << "For how many days do you have sales figures? ";
14     cin >> days;
15
16     // Get the sales for each day and accumulate a total.
17     for (int count = 1; count <= days; count++)
18     {
19         double sales;
20         cout << "Enter the sales for day " << count << ": ";
21         cin >> sales;
22         total += sales; // Accumulate the running total.
23     }
24
```

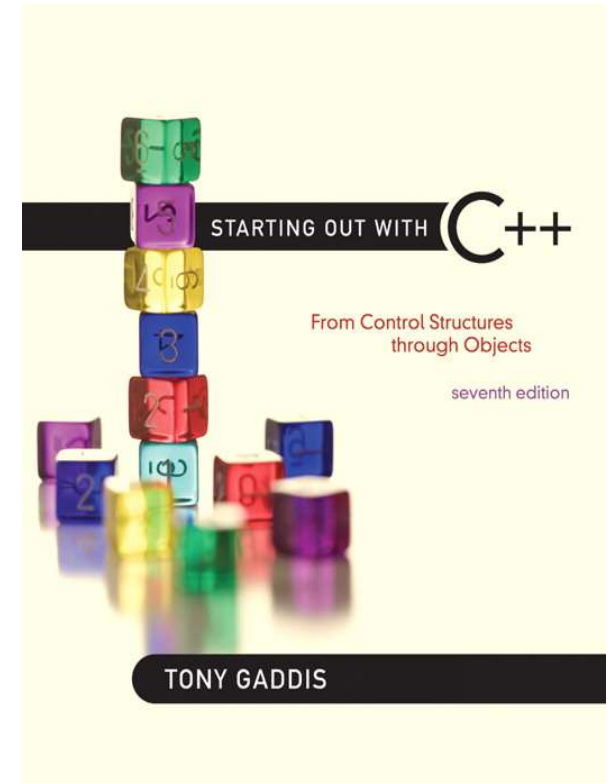
Continued...

# A Running Total in Program 5-12

```
25     // Display the total sales.
26     cout << fixed << showpoint << setprecision(2);
27     cout << "The total sales are $" << total << endl;
28     return 0;
29 }
```

## Program Output with Example Input Shown in Bold

```
For how many days do you have sales figures? 5 [Enter]
Enter the sales for day 1: 489.32 [Enter]
Enter the sales for day 2: 421.65 [Enter]
Enter the sales for day 3: 497.89 [Enter]
Enter the sales for day 4: 532.37 [Enter]
Enter the sales for day 5: 506.92 [Enter]
The total sales are $2448.15
```



# Sentinels



# Sentinels

- sentinel: value in a list of values that indicates end of data
- Special value that cannot be confused with a valid value, *e.g.*, -999 for a test score
- Used to terminate input when user may not know how many values will be entered

# A Sentinel in Program 5-13

## Program 5-13

```
1  // This program calculates the total number of points a
2  // soccer team has earned over a series of games. The user
3  // enters a series of point values, then -1 when finished.
4  #include <iostream>
5  using namespace std;
6
7  int main()
8  {
9      int game = 1,    // Game counter
10         points,      // To hold a number of points
11         total = 0;    // Accumulator
12
13     cout << "Enter the number of points your team has earned\n";
14     cout << "so far in the season, then enter -1 when finished.\n\n";
15     cout << "Enter the points for game " << game << ": ";
16     cin >> points;
17
18     while (points != -1)
19     {
20         total += points;
21         game++;
22         cout << "Enter the points for game " << game << ": ";
23         cin >> points;
24     }
25     cout << "\nThe total points are " << total << endl;
26     return 0;
27 }
```

Continued...

# A Sentinel in Program 5-13

## Program Output with Example Input Shown in Bold

Enter the number of points your team has earned  
so far in the season, then enter -1 when finished.

Enter the points for game 1: **7 [Enter]**  
Enter the points for game 2: **9 [Enter]**  
Enter the points for game 3: **4 [Enter]**  
Enter the points for game 4: **6 [Enter]**  
Enter the points for game 5: **8 [Enter]**  
Enter the points for game 6: **-1 [Enter]**

The total points are 34

# QUESTIONS

THANK YOU