

Ronish Shrestha(19707)  
CS360Lab

1.

```
#include <iostream>
#include <fstream> // pkg is for file processing
#include <iomanip> // for printing format on the monitor
using namespace std;
int GetInches(ifstream& rainFile, int numberOfDays);

int main(){
    float average; // Average rainfall
    float totalRain; // Total accumulated rain
    int numberOfDays; // Number of days in calculation
    ifstream rainFile; // Data file – read from hard drive to memory
    cout << fixed << showpoint;
    rainFile.open("Rain.In");
    rainFile >> numberOfDays;
    totalRain = GetInches(rainFile, numberOfDays);
    if (totalRain == 0.0)
        cout << "There was no rain during this period." << endl;
    else{
        average = totalRain / numberOfDays;
        cout << "The average rain fall over "
        << numberOfDays;
        cout << " days is " << setw(1) << setprecision(3)
        << average << endl;
    }
    return 0;
}

//*****
int GetInches(ifstream& rainFile, int numberOfDays){
    float inches; // Day's worth of rain
    int counter; // Loop control variable
    float totalRain = 0.0;
    counter = 1;
    while (counter <= numberOfDays){
        rainFile >> inches;
        totalRain = totalRain + inches;
        counter++;
    }
    return totalRain;
}
```

```
1 #include <iostream>
2 #include <fstream> // pkg is for file processing
3 #include <iomanip> // for printing format on the monitor
4 using namespace std;
5 int GetInches(ifstream& rainFile, int numberOfDays);
6
7 int main(){
8     float average; // Average rainfall
9     float totalRain; // Total accumulated rain
10    int numberOfDays; // Number of days in calculation
11    ifstream rainFile; // Data file - read from hard drive to memory
12    cout << fixed << showpoint;
13    rainFile.open("Rain.In");
14    rainFile >> numberOfDays;
15    totalRain = GetInches(rainFile, numberOfDays);
16    if (totalRain == 0.0)
17        cout << "There was no rain during this period." << endl;
18    else{
19        average = totalRain / numberOfDays;
20        cout << "The average rain fall over "
21        << numberOfDays;
22        cout << " days is " << setw(1) << setprecision(3)
23        << average << endl;
24    }
25    return 0;
26 }
27 //*****
28 int GetInches(ifstream& rainFile, int numberOfDays){
```

Console output: There was no rain during this period.

2.

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int inches;
```

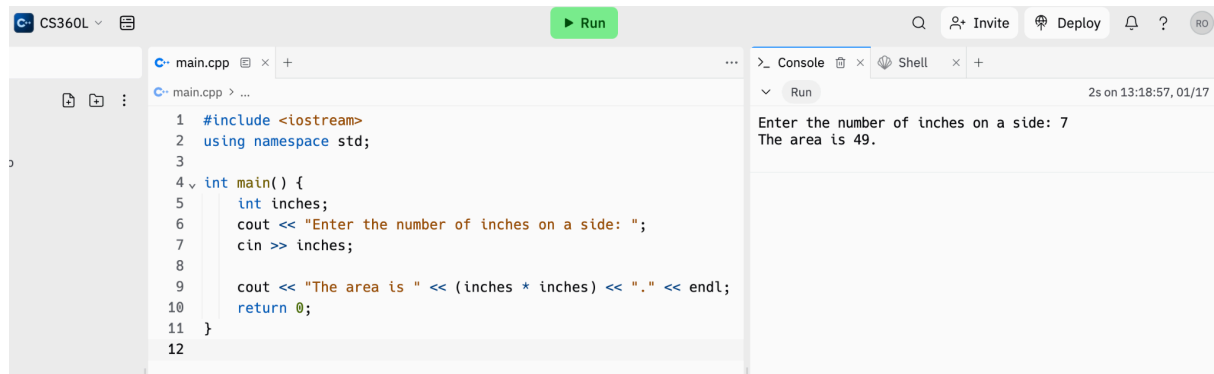
```
    cout << "Enter the number of inches on a side: ";
```

```
    cin >> inches;
```

```
    cout << "The area is " << (inches * inches) << "." << endl;
```

```
    return 0;
```

```
}
```



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int inches;
6     cout << "Enter the number of inches on a side: ";
7     cin >> inches;
8
9     cout << "The area is " << (inches * inches) << "." << endl;
10    return 0;
11 }
12
```

Console output:  
Enter the number of inches on a side: 7  
The area is 49.

`int inches;` declares an integer variable named `inches`.  
`cout << "Enter the number of inches on a side " << endl;` outputs a message prompting the user to enter the side length.  
`cin >> inches;` takes the user's input and stores it in the `inches` variable.  
`cout << endl << "The area is " << inches * inches << "." << endl;` calculates the area by squaring the value of `inches` and outputs the result.  
`return 0;` indicates that the program

3.

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
    int year;
```

```
    // Prompt the user to enter a year  
    cout << "Enter a year AD, for example, 1997: ";  
    cin >> year;
```

```
    // Check if the year is a leap year  
    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {  
        cout << year << " is a leap year." << endl;  
    } else {
```

```

        cout << year << " is not a leap year." << endl;
    }

    return 0;
}

```

The screenshot shows a web-based C++ IDE with the following components:

- File Explorer:** Shows a file named `main.cpp`.
- Code Editor:** Contains the following C++ code:
 

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int year;
7
8      // Prompt the user to enter a year
9      cout << "Enter a year AD, for example, 1997: ";
10     cin >> year;
11
12     // Check if the year is a leap year
13     if ((year % 4 == 0 && year % 100 != 0) || (year % 400 ==
14         0)) {
15         cout << year << " is a leap year." << endl;
16     } else {
17         cout << year << " is not a leap year." << endl;
18     }
19     return 0;
20 }
21

```
- Console:** Shows the output of three test runs:
  - Run 1: "Enter a year AD, for example, 1997: 2002" followed by "2002 is not a leap year."
  - Run 2: "Enter a year AD, for example, 1997: 2020" followed by "2020 is a leap year."
  - Run 3: "Enter a year AD, for example, 1997: 2014" followed by "2014 is not a leap year."

4.

```
#include <iostream>
```

```
int main() {
    int n; // Declare an integer variable to store the number of rows for the kite pattern
```

```

    // Prompt the user to enter the number of rows for the kite pattern
    std::cout << "Enter the number of rows for the kite pattern: ";
    std::cin >> n;

```

```

    // Upper part of the kite
    // Loop over each row for the upper part
    for (int i = 0; i < n; ++i) {
        // Print leading spaces for alignment
        for (int j = 0; j < n - i; ++j) {
            std::cout << " ";
        }
        // Print the kite pattern (asterisks and spaces)
        for (int k = 0; k < 2 * i + 1; ++k) {
            if (k == 0 || k == 2 * i) {

```









```




        std::cout << "**"; // Print an asterisk at the start and end of each line
    } else {
        std::cout << " "; // Fill the middle with spaces
    }
}
std::cout << std::endl; // Move to the next line
}

// Lower part of the kite
// Loop over each row for the lower part (in reverse order)
for (int i = n - 2; i >= 0; --i) {
    // Print leading spaces for alignment
    for (int j = 0; j < n - i; ++j) {
        std::cout << " ";
    }
    // Print the kite pattern (asterisks and spaces)
    for (int k = 0; k < 2 * i + 1; ++k) {
        if (k == 0 || k == 2 * i) {
            std::cout << "**"; // Print an asterisk at the start and end of each line
        } else {
            std::cout << " "; // Fill the middle with spaces
        }
    }
    std::cout << std::endl; // Move to the next line
}

return 0; // Indicate successful program termination
}




```

360L        

main.cpp



```
1 #include <iostream>
2
3 int main() {
4     int n; // Declare an integer variable to store the number
      of rows for the kite pattern
5
6     // Prompt the user to enter the number of rows for the kite
      pattern
7     std::cout << "Enter the number of rows for the kite
      pattern: ";
8     std::cin >> n;
9
10    // Upper part of the kite
11    // Loop over each row for the upper part
12    for (int i = 0; i < n; ++i) {
13        // Print leading spaces for alignment
14        for (int j = 0; j < n - i; ++j) {
15            std::cout << " ";
16        }
17        // Print the kite pattern (asterisks and spaces)
18        for (int k = 0; k < 2 * i + 1; ++k) {
19            if (k == 0 || k == 2 * i) {
20                std::cout << "*"; // Print an asterisk at the
      start and end of each line
21            } else {
22                std::cout << " "; // Fill the middle with spaces
23            }
24        }
25        std::cout << "\n";
26    }
27}
```

Console   

Run 6s on 13:23:20, 01/17

Enter the number of rows for the kite pattern: 7

```
*
 * *
* * *
* * * *
* * * * *
* * * * *
* * * *
* * *
* *
*
```

Core  AI  Ln 48, Col 1 • Spaces: 2 History 