

### 1.3 Console input

#### Console input with prompts

A **console** is a text-based interface for viewing program output and entering user input. One common approach for console-based programs is using prompts for each user input. A **prompt** is a message that indicates what the next expected user input is.

A program using prompts typically uses a cout statement to output the prompt immediately followed by a cin statement.

PARTICIPATION ACTIVITY

1.3.1: Console input with prompts.

Start

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

int main() {
    int goalsFirst;
    int goalsSecond;

    cout << "Enter 1st half goals: ";
    cin >> goalsFirst;
    cout << "Enter 2nd half goals: ";
    cin >> goalsSecond;

    cout << "Your team scored ";
    cout << goalsFirst + goalsSecond;
    cout << " goal(s)." << endl;

    return 0;
}
```

3goalsFirst

5goalsSecond

Enter 1st half goals: 3
Enter 2nd half goals: 5
Your team scored 8 goal(s).
|

Captions

Feedback?

PARTICIPATION ACTIVITY

1.3.2: Console input with prompts.

1) A program will \_\_\_\_ while executing a cin statement.

time out after about thirty seconds and then continue to the next statement

wait briefly and then continue to the next statement

wait indefinitely

2) The prompt for a value may be placed before or after a cin statement.

True

False

3) More than one cout statement may be used to print a prompt.

True

False

Feedback?

#### Console input without prompts

Another common approach for console-based programs is to read user input without using prompts. In this approach, a user typically enters multiple expected inputs with each input value separated by spaces or tab characters.

PARTICIPATION ACTIVITY

1.3.3: Console input without prompts.

Start

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

int main() {
    int goalsFirst;
    int goalsSecond;

    cin >> goalsFirst;
    cin >> goalsSecond;

    cout << "Your team scored ";
    cout << goalsFirst + goalsSecond;
    cout << " goal(s)." << endl;

    return 0;
}
```

3goalsFirst

5goalsSecond

3 5
Your team scored 8 goal(s).
|

Captions

Feedback?

PARTICIPATION ACTIVITY

1.3.4: Console input without prompts.

1) For a program without prompts, expected program inputs must be entered all at once.

True

False

2) What is an advantage of entering expected program inputs all at once?

Input error checking is improved.

User interaction is faster.

3) What happens if more values are entered all at once than there are cin statements?

The extra values are ignored.

The program will stop with a runtime error.

The program will wait for the user to enter more cin statements.

Feedback?

#### Pre-entered input with zyBooks Development Environment (zyDE)

The zyBooks development environment (zyDE) uses an input box for program input, in which a user can pre-enter program input before the program is run. The input box appears in the upper right corner of the zyDE.

To run a program in the zyDE, enter all program inputs and click the Run button. Each cin statement in the program reads the next value from the pre-entered input.

When using pre-entered input in a zyDE, the input values are not displayed inline with the program output. The zyDE will only display the program output, which behaves differently from executing a program interactively in a console.

The following zyDEs show the same program with prompts and without prompts. When using pre-entered input in a zyDE, the input values are not displayed inline with the program output. The zyDE will only display the program output, which behaves differently from executing a program interactively in a console. Try running each program with the same input, and observe the differences in the output.

zyDE 1.3.1: Program with prompts.

Load default template...

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int goalsFirst;
6     int goalsSecond;
7
8     cout << "Enter 1st half goals: ";
9     cin >> goalsFirst;
10    cout << "Enter 2nd half goals: ";
11    cin >> goalsSecond;
12
13    cout << "Your team scored ";
14    cout << goalsFirst + goalsSecond;
15    cout << " goal(s)." << endl;
16
17    return 0;
}
```

3 5

Run

Feedback?

zyDE 1.3.2: Program without prompts.

Load default template...

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int goalsFirst;
6     int goalsSecond;
7
8     cin >> goalsFirst;
9     cin >> goalsSecond;
10
11    cout << "Your team scored ";
12    cout << goalsFirst + goalsSecond;
13    cout << " goal(s)." << endl;
14
15    return 0;
16 }
17
```

3 5

Run

Feedback?

PARTICIPATION ACTIVITY

1.3.5: zyDE input box.

1) When are values entered in the zyDE input box?

While the program is running.

After the program is run.

Before a program is run.

2) How are values in the zyDE input box processed?

For each cin statement, the next value is read from the zyDE input box. The value is removed from the input box.

For each cin statement, the next value is read from the zyDE input box. The value remains in the input box.

For each cin statement, the next line is read from the zyDE input box.

Feedback?