Tishk International University
Faculty of Applied Science
Information Technology Department

Expressions and Interactivity

Lecture 4

Fall 2025

Course Code: IT117

Grade 1

Islam Abdulazeez
islam.abdulaziz@tiu.edu.iq

December 21, 2025

Programming I

Outlines

- ✓ User input using cin and getline()
- ✓ Arithmetic operators and expressions
- ✓ Operator precedence and evaluation
- ✓ Type casting and division behavior
- ✓ ASCII codes, random numbers, and flowcharts

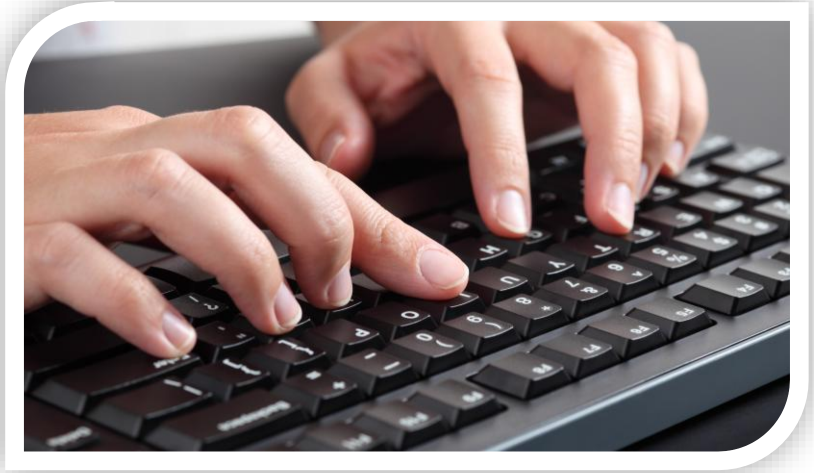
Learning Outcomes

- **At the end of today’s session, you will be able to:**

  - ✓ Explain user input in C++ using cin and getline().
  - ✓ Apply arithmetic operators and operator precedence correctly.
  - ✓ Analyze division results using type casting.
  - ✓ Create C++ programs using expressions and input.

User Input

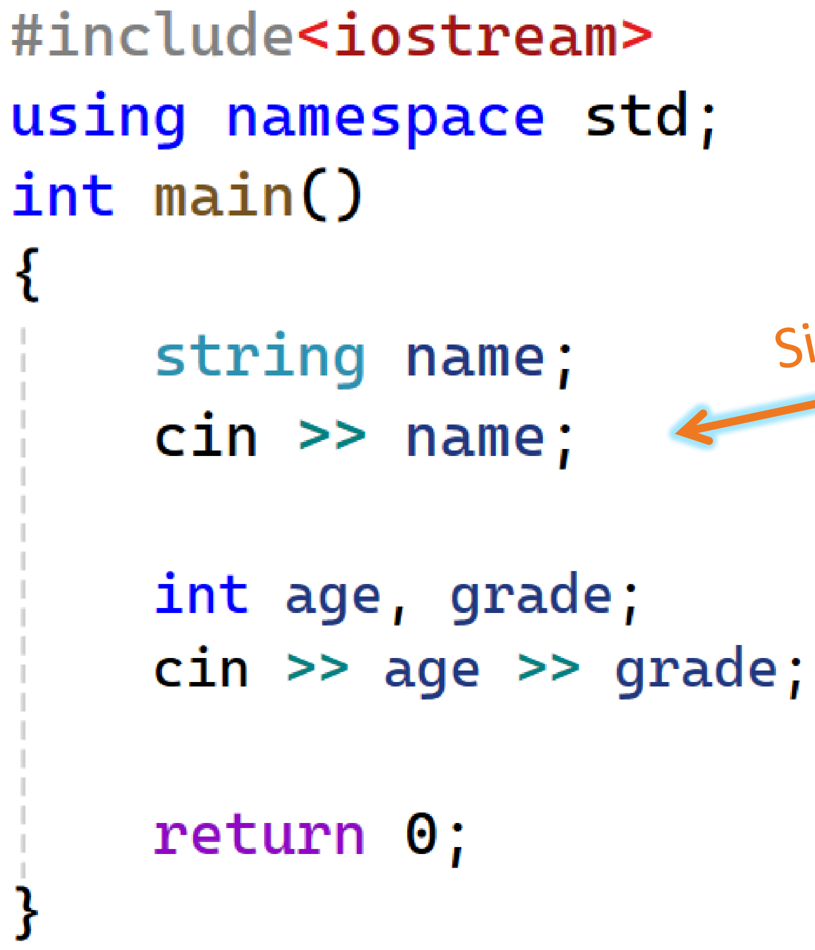
- cin is an object in C++ used for reading data from the standard input stream.
- It is part of the iostream library.
- The extraction operator >> is used with cin.
- Input is stored in variables based on their data types.
- Execution waits until the user provides input.



Syntax

```
// Single Input
cin >> variable;

// Multiple Inputs
cin >> variable >> variable ;
```

User Input (Ex.)



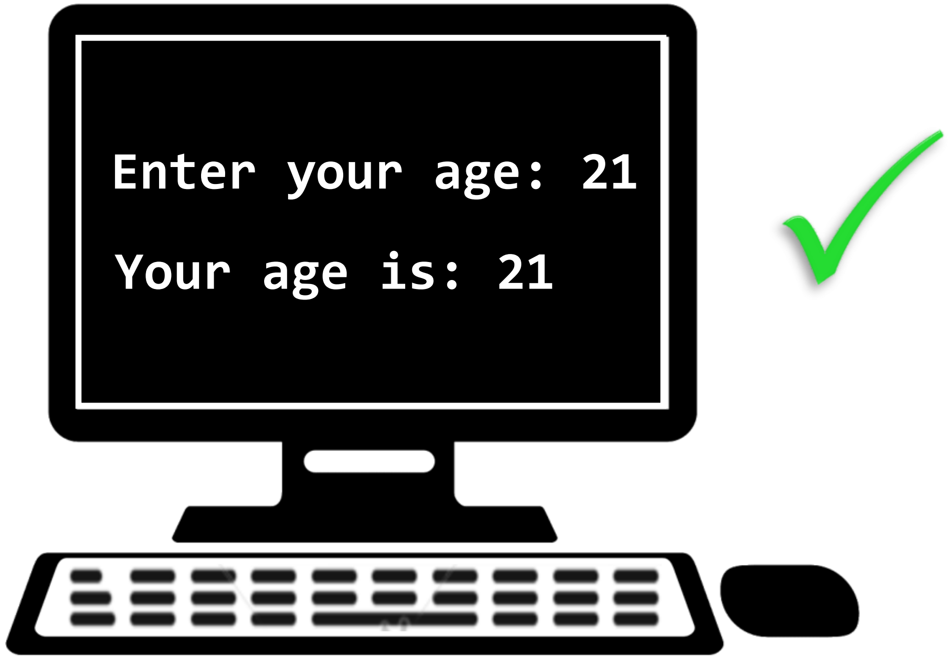
```
#include<iostream>
using namespace std;
int main()
{
    string name;
    cin >> name;

    int age, grade;
    cin >> age >> grade;

    return 0;
}
```

Single Input

Multiple Inputs

User Input (Ex.)

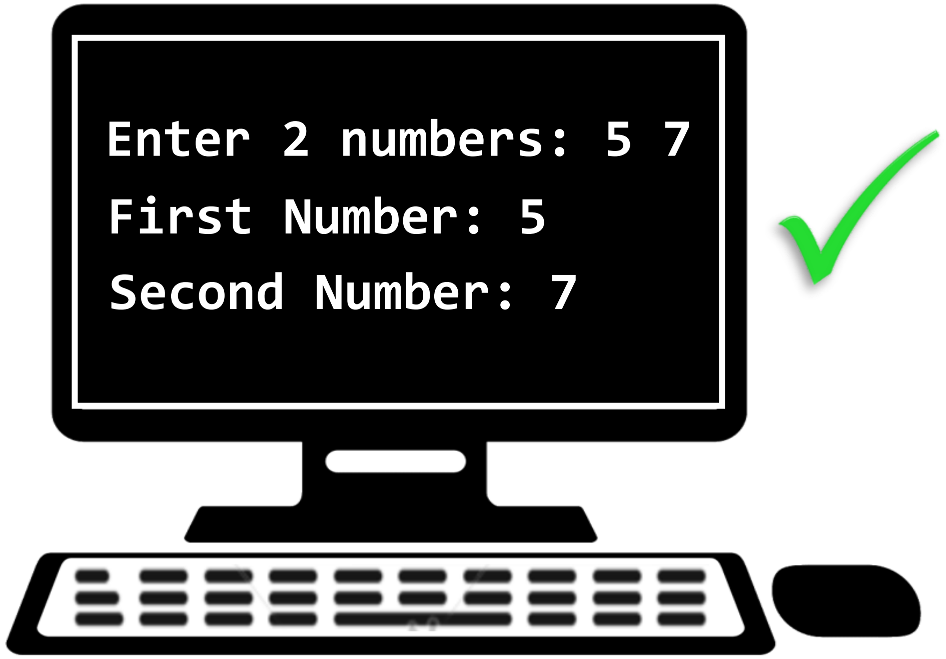
```cpp
#include <iostream>
using namespace std;
int main() {

    int age;
    cout << "Enter your age: ";
    cin >> age;
    cout << "Your age is: " << age;

    return 0;
}
```



```
Enter your age: 21

Your age is: 21
```

User Input (Ex.)

```cpp
#include<iostream>
using namespace std;
int main()
{
    int num1, num2;
    cout << "Enter 2 numbers: ";
    cin >> num1 >> num2;
    cout << "First number: " << num1 << endl;
    cout << "Second number: " << num2 << endl;

    return 0;
}
```



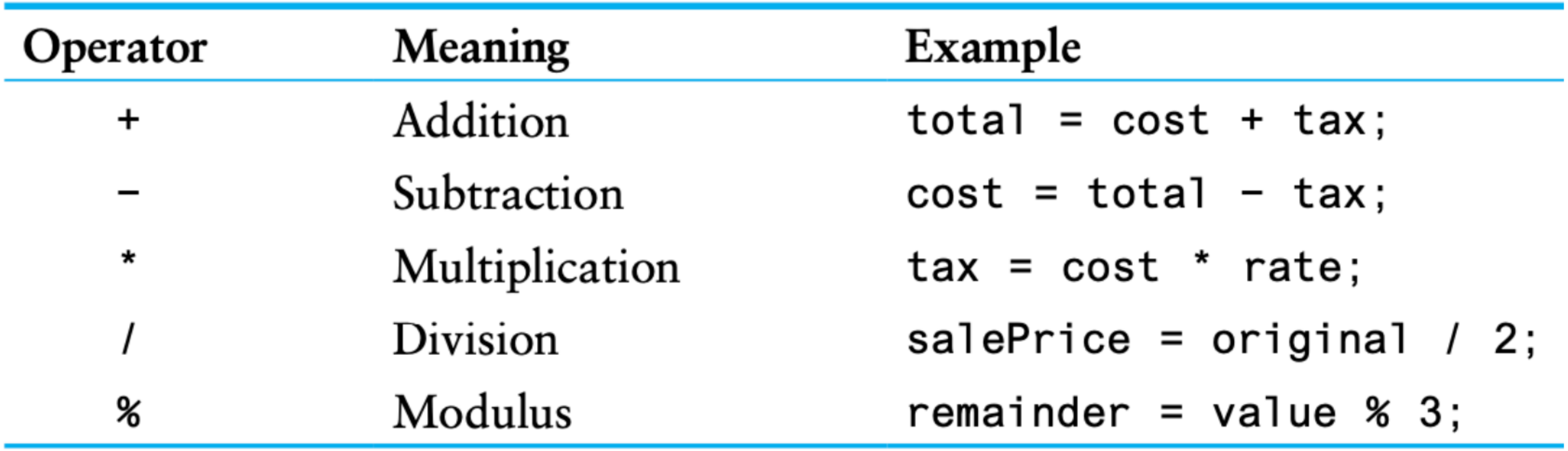
```
Enter 2 numbers: 5 7
First Number: 5
Second Number: 7
```

User Input (Ex.)

```
#include<iostream>
using namespace std;
int main()
{
    string firstName;
    int age;

    cout << "Enter your first name:  ";
    cin >> firstName;  // Lawin
    cout << "Enter your age:  ";
    cin >> age;  //20

    cout << "Your name is " << firstName << " and you are " << age << "years old.";

    return 0;
}
```

Output -> Your name is Lawin and you are 20 years old.

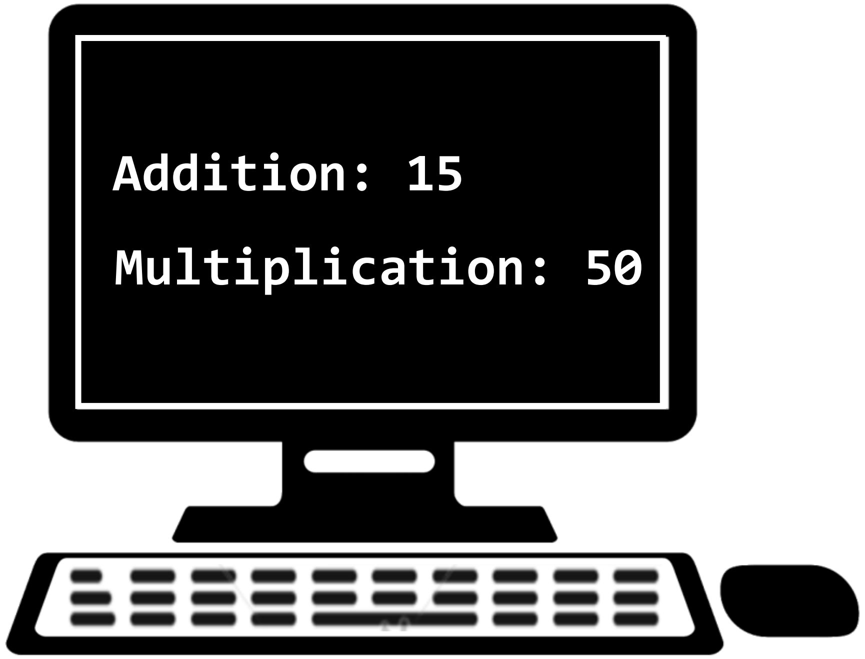
Arithmetic Operators



Operator	Meaning	Example
`+`	Addition	`total = cost + tax;`
`–`	Subtraction	`cost = total – tax;`
`*`	Multiplication	`tax = cost * rate;`
`/`	Division	`salePrice = original / 2;`
`%`	Modulus	`remainder = value % 3;`

Arithmetic Operators (Ex.)

```cpp
#include<iostream>
using namespace std;
int main()
{
    int num1 = 10, num2 = 5;

    cout << "Addition: " << num1 + num2 << endl;
    cout << "Multiplication: " << num1 * num2;

    return 0;
}
```



```
Addition: 15

Multiplication: 50
```

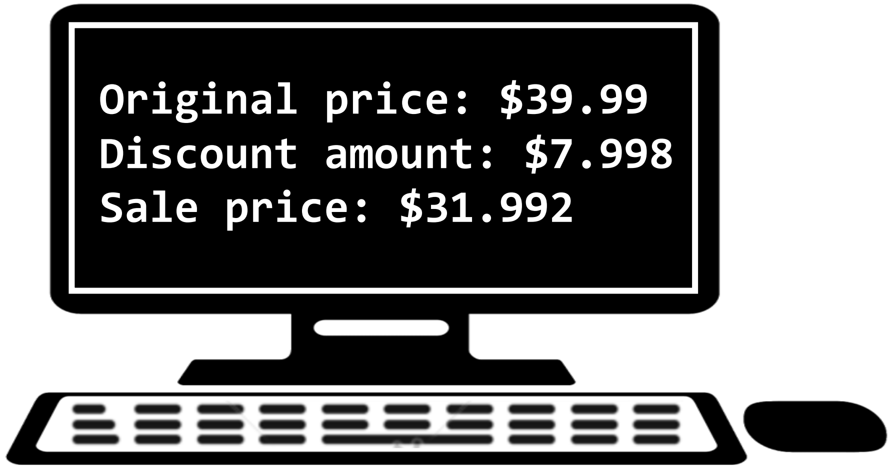
Arithmetic Operators (Ex.)

```
#include<iostream>
using namespace std;
int main()
{
    double price = 17.99;
    int quantity = 2;

    double total = price * quantity;
    cout << "Total: " << total;

    return 0;
}
```

```
Total: 35.98
```

Arithmetic Operators (Ex.)

Q. Create a C++ program to calculate the sale price of an item originally priced at $39.99 with a 20% discount. The program should display the original price, discount amount, and final sale price.



```
Original price: $39.99
Discount amount: $7.998
Sale price: $31.992
```

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

int main() {

    double originalPrice = 39.99;
    double discount;
    double salePrice;

    // Calculate 20% discount
    discount = originalPrice * 0.20;

    // Calculate sale price
    salePrice = originalPrice - discount;

    // Display results
    cout << "Original price: $" << originalPrice << endl;
    cout << "Discount amount: $" << discount << endl;
    cout << "Sale price: $" << salePrice << endl;

    return 0;
}
```

Precedence(Priority) of Operators

$$6 / 2 (1 + 2) = 9$$ ✓

Precedence(Priority) of Operators

Precedence of Arithmetic Operators:

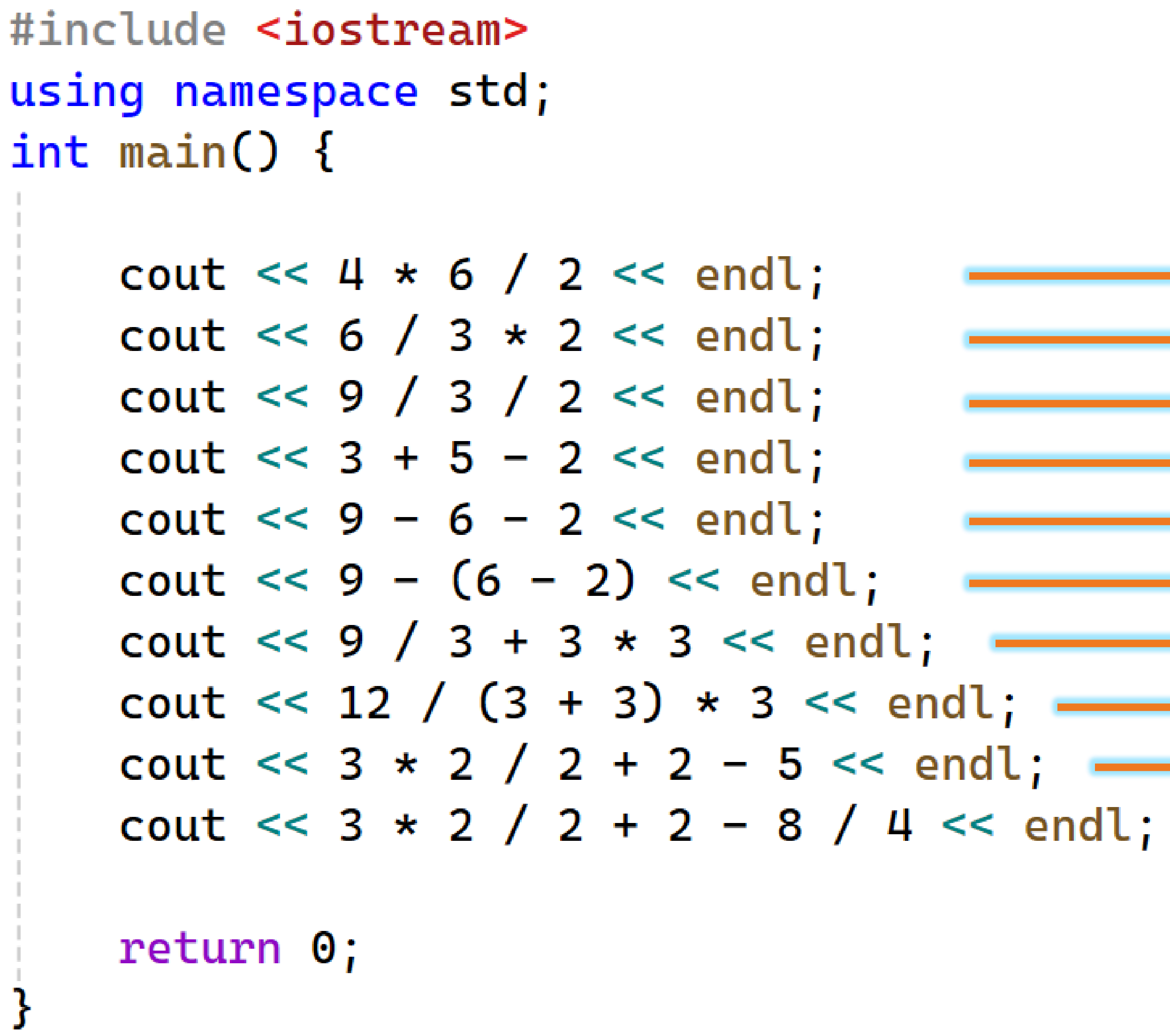
- Parentheses ( ) are evaluated first. The expression in the innermost parentheses is evaluated first if the parentheses are nested.
- Multiplication (*), division (/), and modulus (%), all at the same precedence and evaluated left to right.
- Addition (+) and subtraction (-) are evaluated last, have the same precedence, and are evaluated from left to right.

```
3 * 5 + 2     = 17
3 * (5 + 2)   = 21
5 + 3 * 4 – 2 = 15
6 * 8 / 4     = 12
6 * (8 / 4)   = 12
```

```
(12 / (8 – 2))  = 2
8 + (7 - 9)     = 6
9 + 3 + 4 – 2   = 14
16 / 4 * 2      = 8
4 / 2 - 2       = 0
```

- In **C++**, it's the same.

Precedence(Priority) of Operators (Ex.)



```cpp
#include <iostream>
using namespace std;
int main() {

    cout << 4 * 6 / 2 << endl;
    cout << 6 / 3 * 2 << endl;
    cout << 9 / 3 / 2 << endl;
    cout << 3 + 5 - 2 << endl;
    cout << 9 - 6 - 2 << endl;
    cout << 9 - (6 - 2) << endl;
    cout << 9 / 3 + 3 * 3 << endl;
    cout << 12 / (3 + 3) * 3 << endl;
    cout << 3 * 2 / 2 + 2 - 5 << endl;
    cout << 3 * 2 / 2 + 2 - 8 / 4 << endl;

    return 0;
}
```

```
12
4
1
6
1
5
12
6
0
3
```

Arithmetic Expressions in C++

- Arithmetic expressions in C++ must be entered into the computer in straight line form.

$$\frac{A + B}{C + D}$$ In C++ is written as `(A + B)  /  (C + D)`

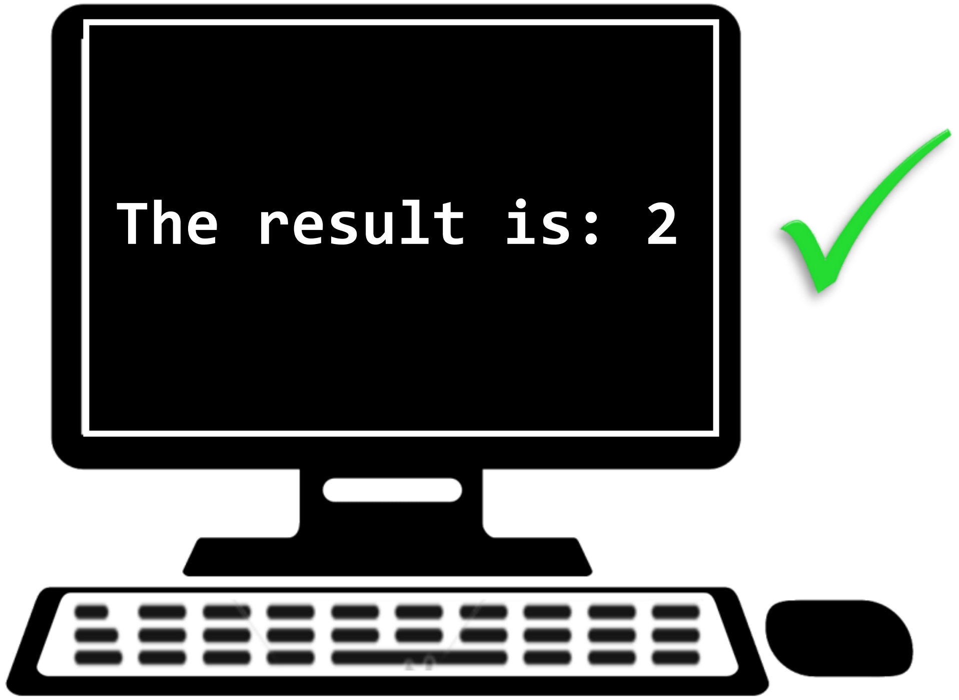
$$A + \frac{B}{C} + D$$ In C++ is written as `A + (B / C) + D`

$$\frac{1}{1 + 2x^2}$$ In C++ is written as `1 / (1 + 2 * x * x)`

Arithmetic Expressions in C++ (Ex.)

$$\frac{A + B}{C + D}$$

```cpp
#include <iostream>
using namespace std;
int main() {

    int A = 12, B = 8, C = 7, D = 3;
    int result = (A + B) / (C + D);
    cout << "The result is: " << result;

    return 0;
}
```

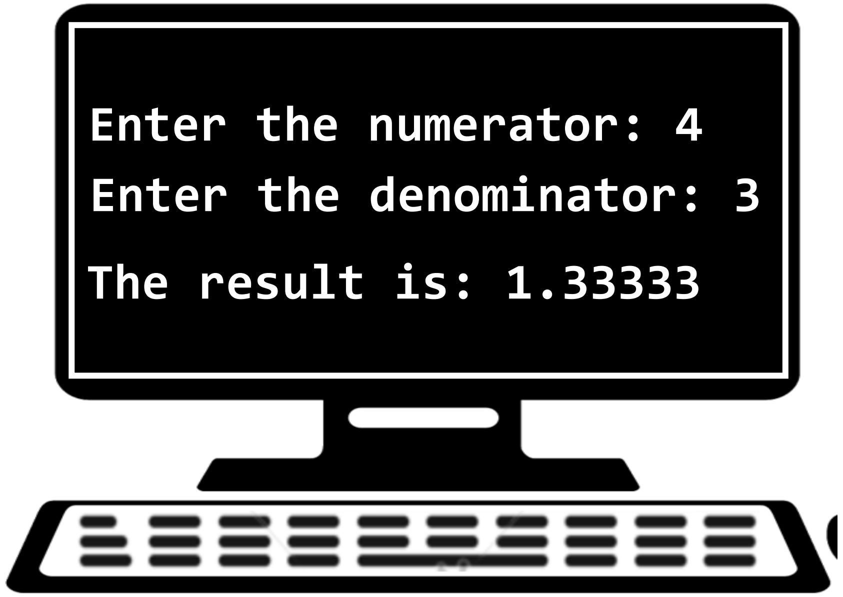


```
The result is: 2
```

Mathematical Expressions - Division

- If one (or both) of the operands of the division operator is double, the result will be double.

```cpp
#include <iostream>
using namespace std;
int main() {

    double numerator;
    int denominator;

    cout << "Enter the numerator: ";
    cin >> numerator;
    cout << "Enter the denominator: ";
    cin >> denominator;

    cout << "The result is: " << numerator / denominator;
    return 0;
}
```

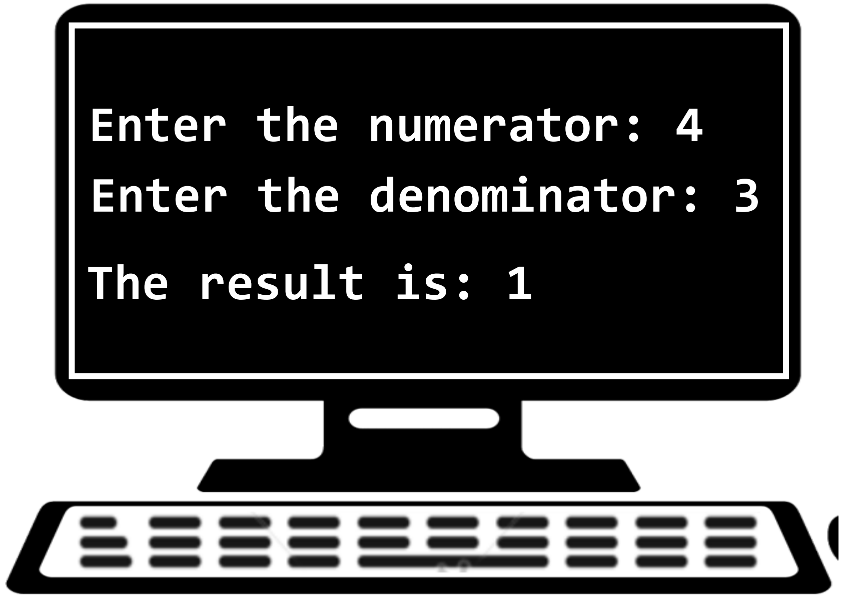


```
Enter the numerator: 4
Enter the denominator: 3
The result is: 1.33333
```

Mathematical Expressions – Division

- If the operands of the division operator are both integers, the result will be an integer.

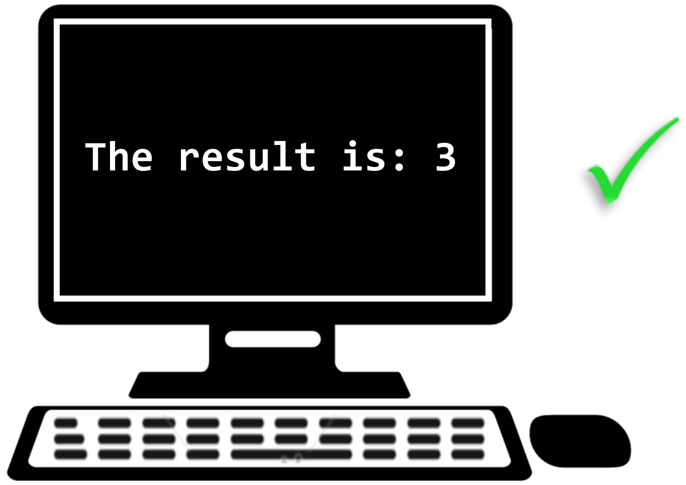
```cpp
#include <iostream>
using namespace std;
int main() {

    int numerator, denominator;

    cout << "Enter the numerator: ";
    cin >> numerator;
    cout << "Enter the denominator: ";
    cin >> denominator;

    cout << "The result is: " << numerator / denominator;
    return 0;
}
```



```
Enter the numerator: 4
Enter the denominator: 3
The result is: 1
```

Mathematical Expressions – Division (Ex.)

```
#include <iostream>
using namespace std;
int main() {

    int x=10, y=3;
    double result = x / y;

    cout << "The result is: " << result;
    return 0;
}
```

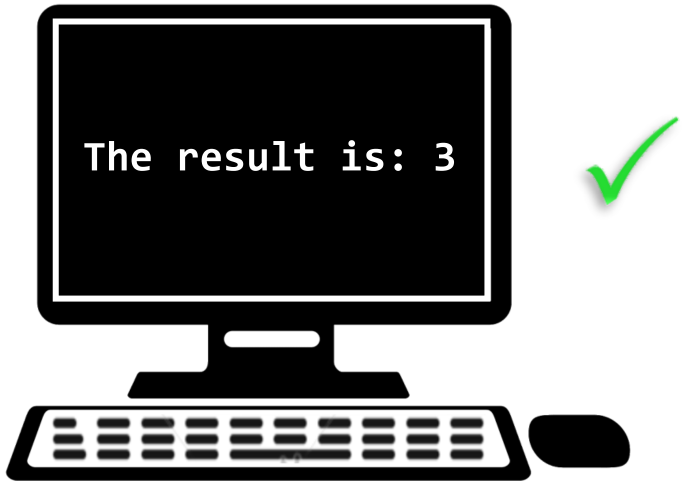


```
The result is: 3
```

Mathematical Expressions – Division (Ex.)

```cpp
#include <iostream>
using namespace std;
int main() {

    double x=10, y=3;
    int result = x / y;

    cout << "The result is: " << result;
    return 0;
}
```

?!



```
The result is: 3
```

Type Casting

- Used to convert one data type to another.

The general format of a type cast expression is:

Data_type(Value)

Type Casting

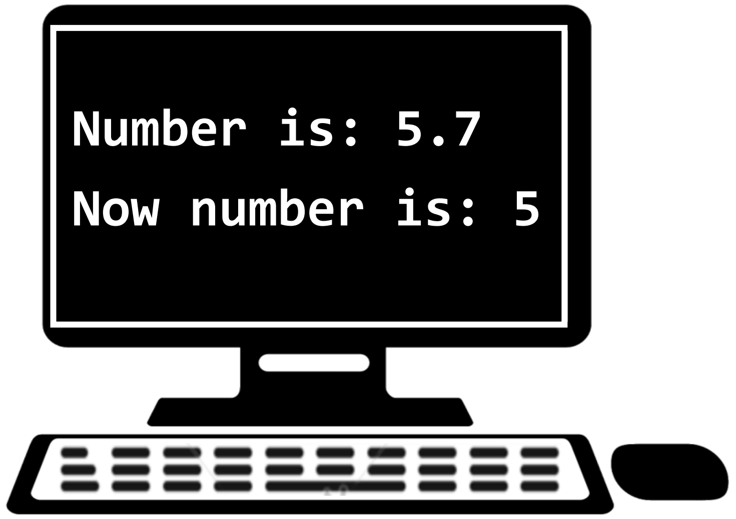
- A value of any built-in type can be converted to any other built-in type

Example

```
int(3.14)        // Converts 3.14 to int, resulting in 3
double(2)      // Converts 2 to a double value (2.0)
char(65)       // Converts 65 to the character A
```

Type Casting (Ex.)

```cpp
#include <iostream>
using namespace std;
int main() {

    double num = 5.7;
    cout << "Number is: " << num << endl;
    cout << "Now number is: " << int(num) << endl;

    return 0;
}
```



```
Number is: 5.7
Now number is: 5
```

Type Casting (Ex.)

- Write a C++ program that asks the user for the number of books they plan to read and the number of months it will take, then calculates and displays the average number of books read per month.

```
// Declare variables
int books;
int months;
double booksPerMonth;

// Get user input
cout << "How many books do you plan to read? ";
cin >> books;  // 15

cout << "How many months will it take you to read them? ";
cin >> months;  // 6

// Calculate average books per month
booksPerMonth = double(books) / months;

// Display result
cout << "That is " << booksPerMonth << " books per month." << endl;
```

Output

```
That is 2.5 books per month.
```

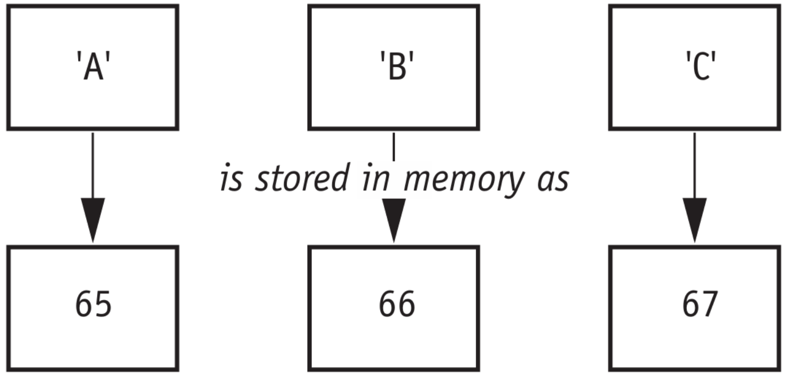
Arithmetic Operators and Expressions (Ex.)

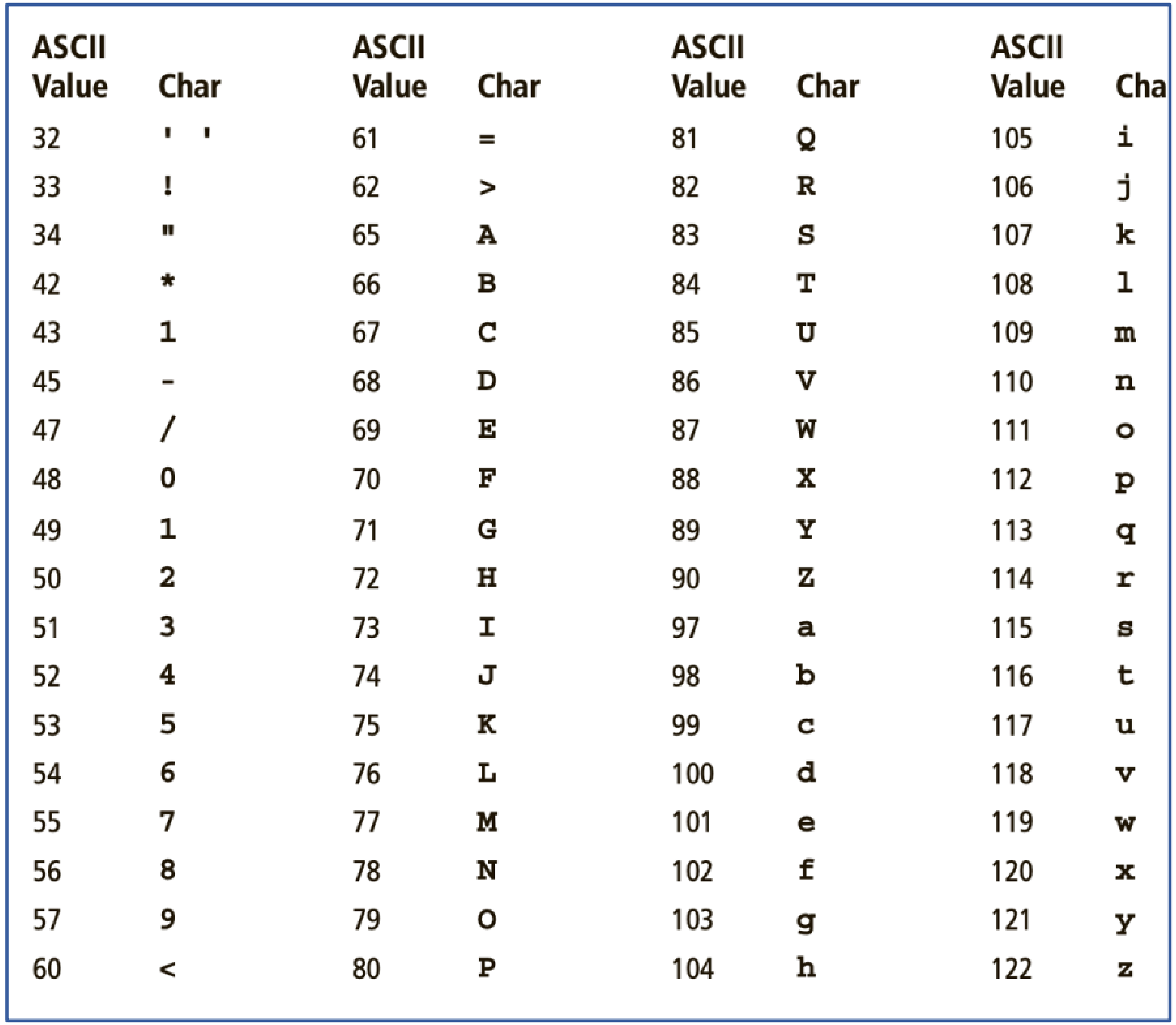
```
cout << "11 / 5 = " << 11 / 5 << "\n";
cout << "11.0 / 5.0 = " << 11.0 / 5.0 << "\n";

cout << "int(11.0) / 5.0 = " << int(11.0) / 5.0 << "\n";
cout << "int(11.0 / 5.0) = " << int(11.0 / 5.0) << "\n";
cout << "float(11 / 5) = " << float(11 / 5) << "\n";
cout << "float(11) / 5 = " << float(11) / 5 << "\n";

cout << "10 / 3.0 = " << 10 / 3.0 << "\n";

cout << "10 % 3 = " << 10 % 3 << "\n";
```

```
11 / 5 = 2
11.0 / 5.0 = 2.2

int(11.0) / 5.0 = 2.2
Int(11.0 / 5.0) = 2
float(11 / 5) = 2
float(11) / 5 = 2.2

10 / 3.0 = 3.33333

10 % 3 = 1
```

ASCII (Character Encoding)

- **ASCII**, which stands for **American Standard Code for Information Interchange**, is a widely used character encoding standard and forms the basis of modern encodings such as Unicode.



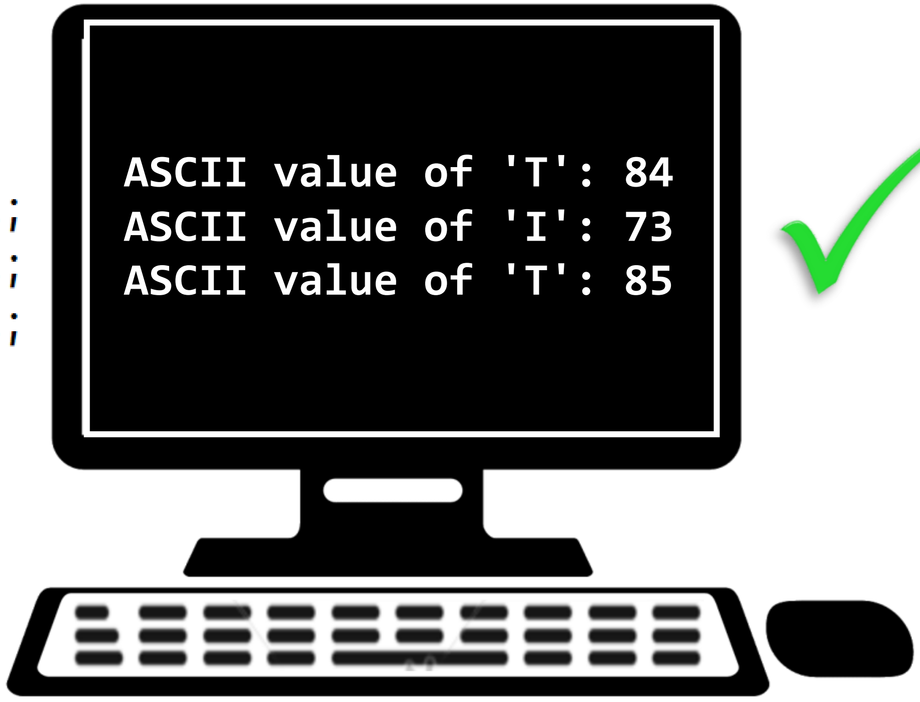


ASCII Value	Char	ASCII Value	Char	ASCII Value	Char	ASCII Value	Cha
32	' '	61	=	81	Q	105	i
33	!	62	>	82	R	106	j
34	"	65	A	83	S	107	k
42	*	66	B	84	T	108	l
43	1	67	C	85	U	109	m
45	-	68	D	86	V	110	n
47	/	69	E	87	W	111	o
48	0	70	F	88	X	112	p
49	1	71	G	89	Y	113	q
50	2	72	H	90	Z	114	r
51	3	73	I	97	a	115	s
52	4	74	J	98	b	116	t
53	5	75	K	99	c	117	u
54	6	76	L	100	d	118	v
55	7	77	M	101	e	119	w
56	8	78	N	102	f	120	x
57	9	79	O	103	g	121	y
60	<	80	P	104	h	122	z

ASCII (Character Encoding)

Q. Write a program that is printing the corresponding ASCII of each letter of "TIU"

```cpp
#include <iostream>
using namespace std;
int main() {

    cout << "ASCII value of 'T': " << int('T') << endl;
    cout << "ASCII value of 'I': " << int('I') << endl;
    cout << "ASCII value of 'U': " << int('U') << endl;

    return 0;
}
```

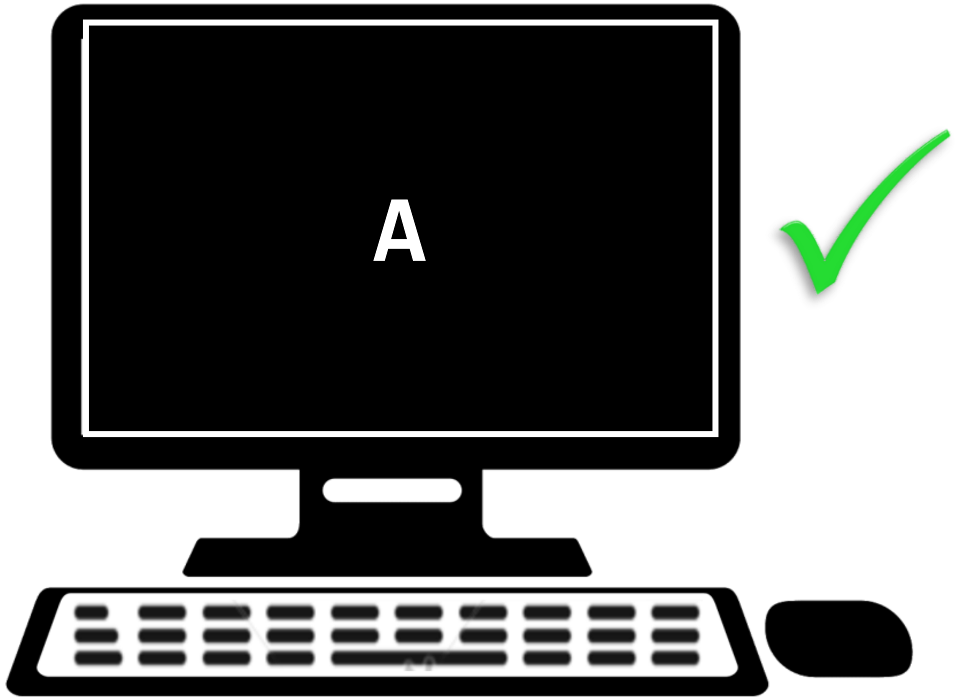


```
ASCII value of 'T': 84
ASCII value of 'I': 73
ASCII value of 'T': 85
```

ASCII (Character Encoding)

Example

```
#include <iostream>
using namespace std;
int main() {

    char asciiCode = 65;
    cout << asciiCode;

    return 0;
}
```



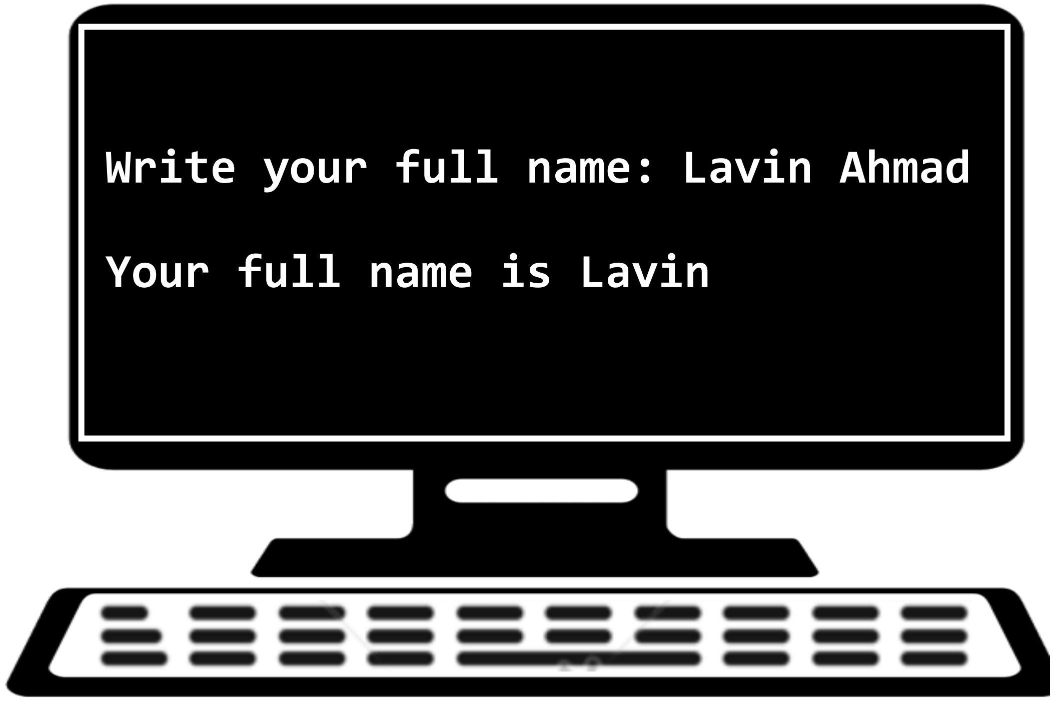
Input Strings Using getline()

- Using cin with the >> operator for string input can lead to potential issues.
- The >> operator skips leading whitespace characters (such as **spaces**, **tabs**, and **line breaks**).
- It begins reading at the first non-whitespace character and **stops** reading when it encounters the next whitespace character.
- To address this limitation, the **getline()** function in C++ can be used.
- **getline()** reads an entire line of input, including leading and embedded spaces, and stores it in a string object.

Input Strings Using getline()

- Using cin

```cpp
#include <iostream>
using namespace std;
int main() {

    string fullName;
    cout << "Wite your full name: ";
    cin >> fullName;
    cout << "Your full name is: " << fullName;

    return 0;
}
```



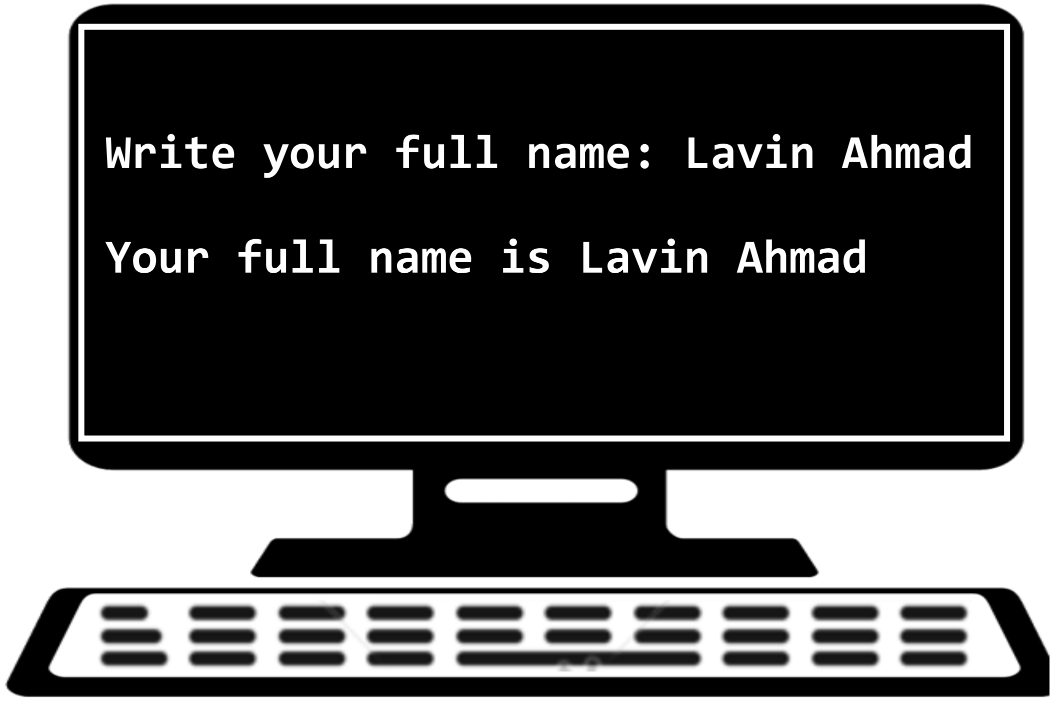
```
Write your full name: Lavin Ahmad

Your full name is Lavin
```

Input Strings Using getline() (Ex.)

- **Using getline**

```
#include <iostream>
#include <string>
using namespace std;
int main() {

    string fullName;
    cout << "Wite your full name: ";
    getline(cin, fullName);
    cout << "Your full name is: " << fullName;

    return 0;
}
```

Add string library

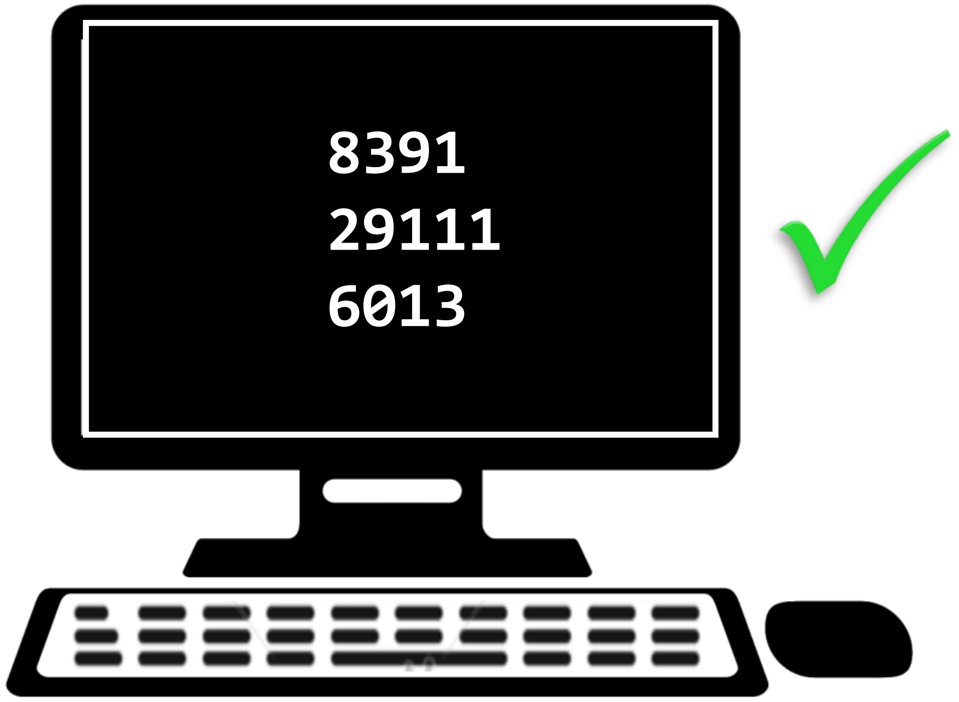


```
Write your full name: Lavin Ahmad

Your full name is Lavin Ahmad
```

Random Numbers

- Random numbers are useful for many programming tasks. The C++ library provides a function, **rand()**, that can be used to generate random numbers.

```
#include <iostream>
using namespace std;
int main() {

    // Seed the random number generator
    srand(time(0));

    // Display three random numbers
    cout << rand() << endl;
    cout << rand() << endl;
    cout << rand() << endl;

    return 0;
}
```



```
8391
29111
6013
```

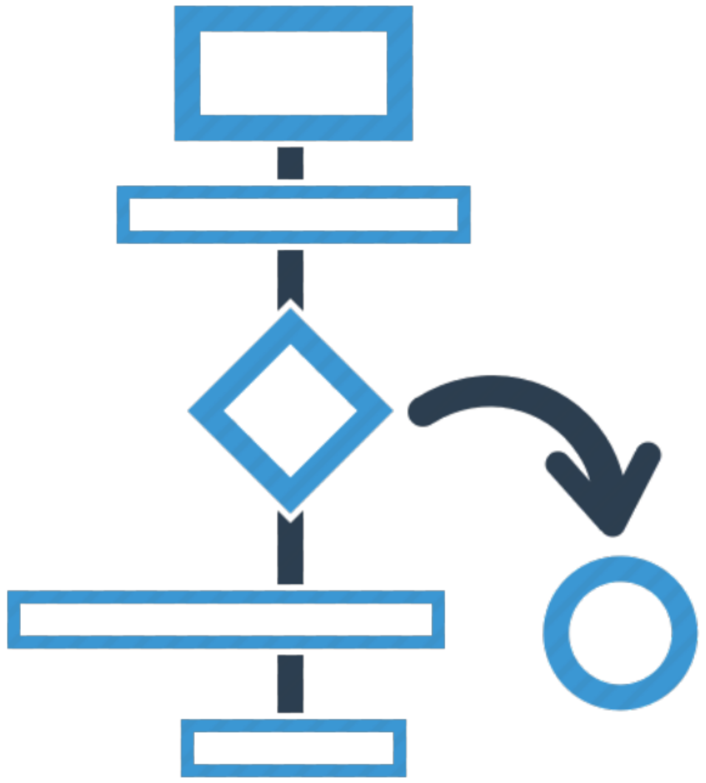
Random Numbers

- Generate Random Numbers in a specific ranges.

```cpp
#include <iostream>
using namespace std;
int main() {

    srand(time(0));

    cout << "Random number (0-99): " << rand() % 100 << endl;
    cout << "Random number (1-100): " << rand() % 100 + 1 << endl;
    cout << "Random number (10-50): " << rand() % 41 + 10;

    return 0;
}
```

Flowchart

- A flowchart is a picture (graphical representation) of the problem solving process.
- A flowchart gives a step-by-step procedure for solution of a problem.
- Using flowcharts can show the sequence and logic of each step before writing a computer program.
- Even people with little programming knowledge can understand flowcharts.

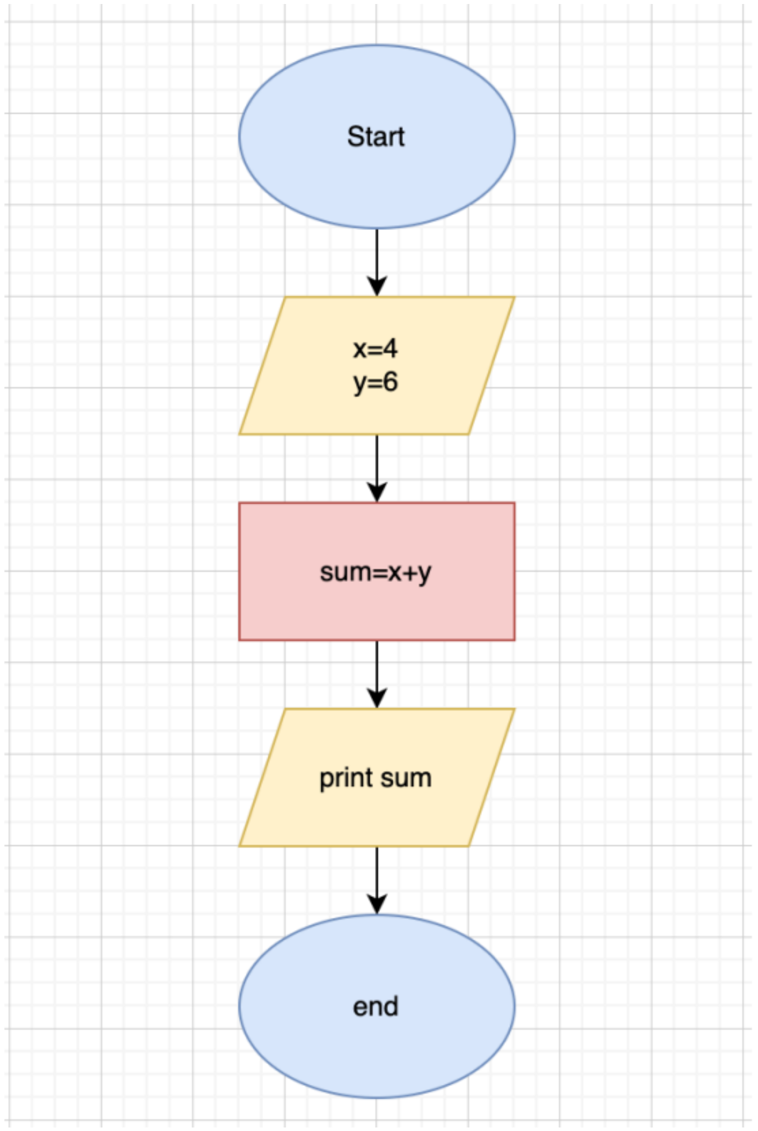


Flowchart

	Symbol	Symbol Name	Purpose
Ellipse	⬭	Start/Stop	Used at the beginning and end of the algorithm to show start and end of the program.
Rectangle	▭	Process	Indicates processes like mathematical operations.
Parallelogram	▱	Input/ Output	Used for denoting program inputs and outputs.
Diamond	◇	Decision	Stands for decision statements in a program, where answer is usually Yes or No.
	↓	Arrow	Shows relationships between different shapes.

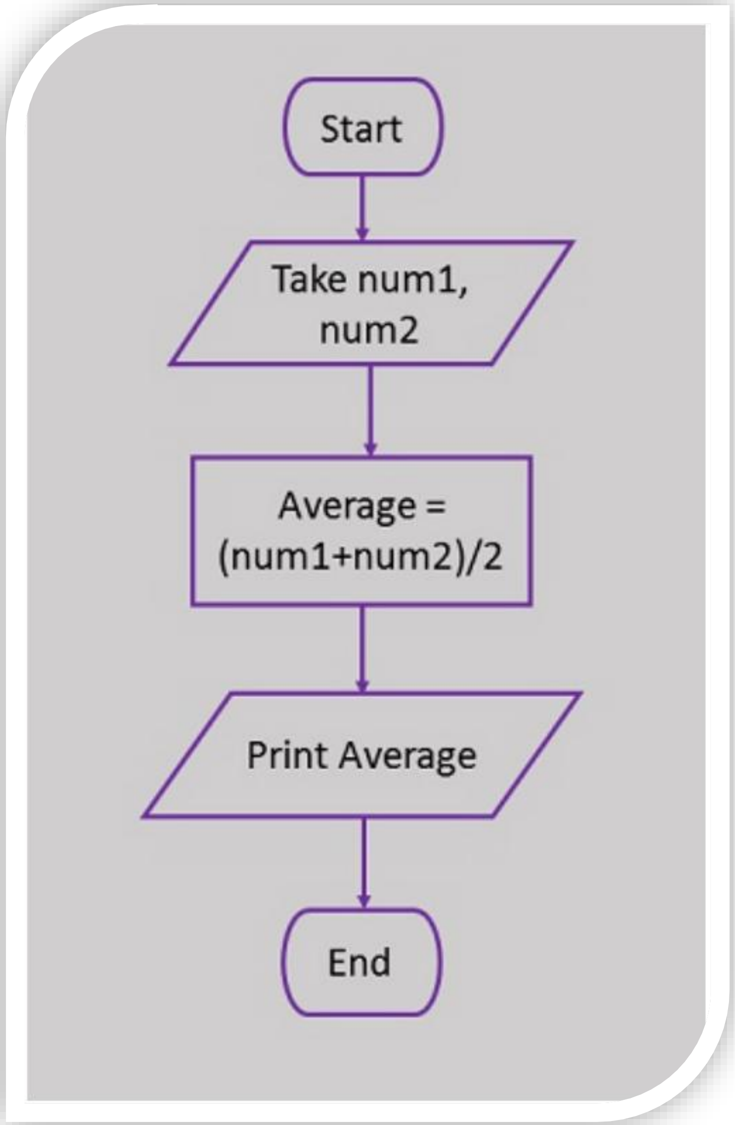
Flowchart (Ex.)

- Draw a flowchart to represent the process of calculating the summation of two numbers.



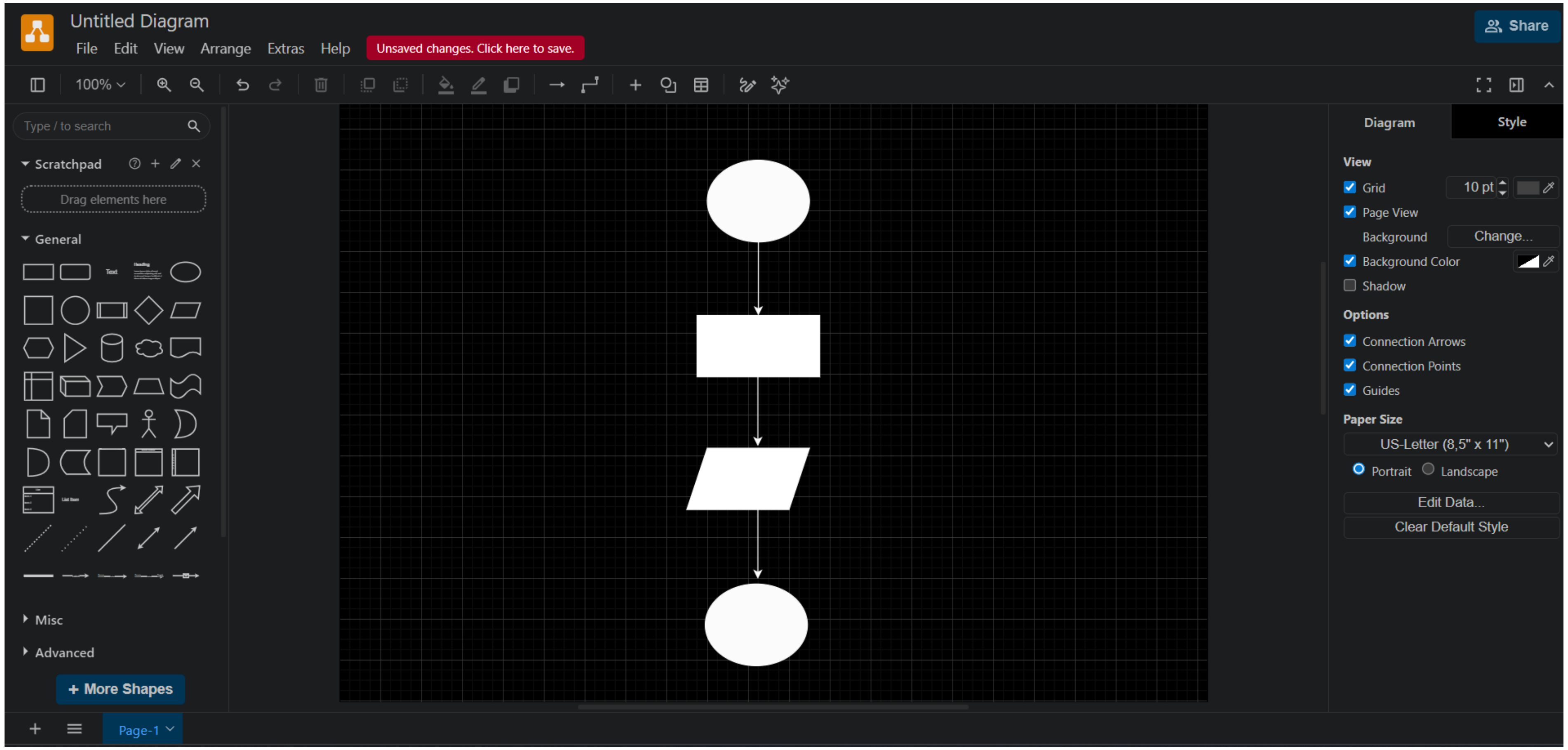
Flowchart (Ex.)

- Draw a flowchart to represent the process of calculating the average of two numbers.



Flowchart

- **draw.io** is a popular online diagramming tool that allows users to create various types of diagrams, including flowcharts, accessible via draw.io.



View

- [✓] Grid 10 pt
- [✓] Page View
- Background Change...
- [✓] Background Color
- [ ] Shadow

Options

- [✓] Connection Arrows
- [✓] Connection Points
- [✓] Guides

Paper Size

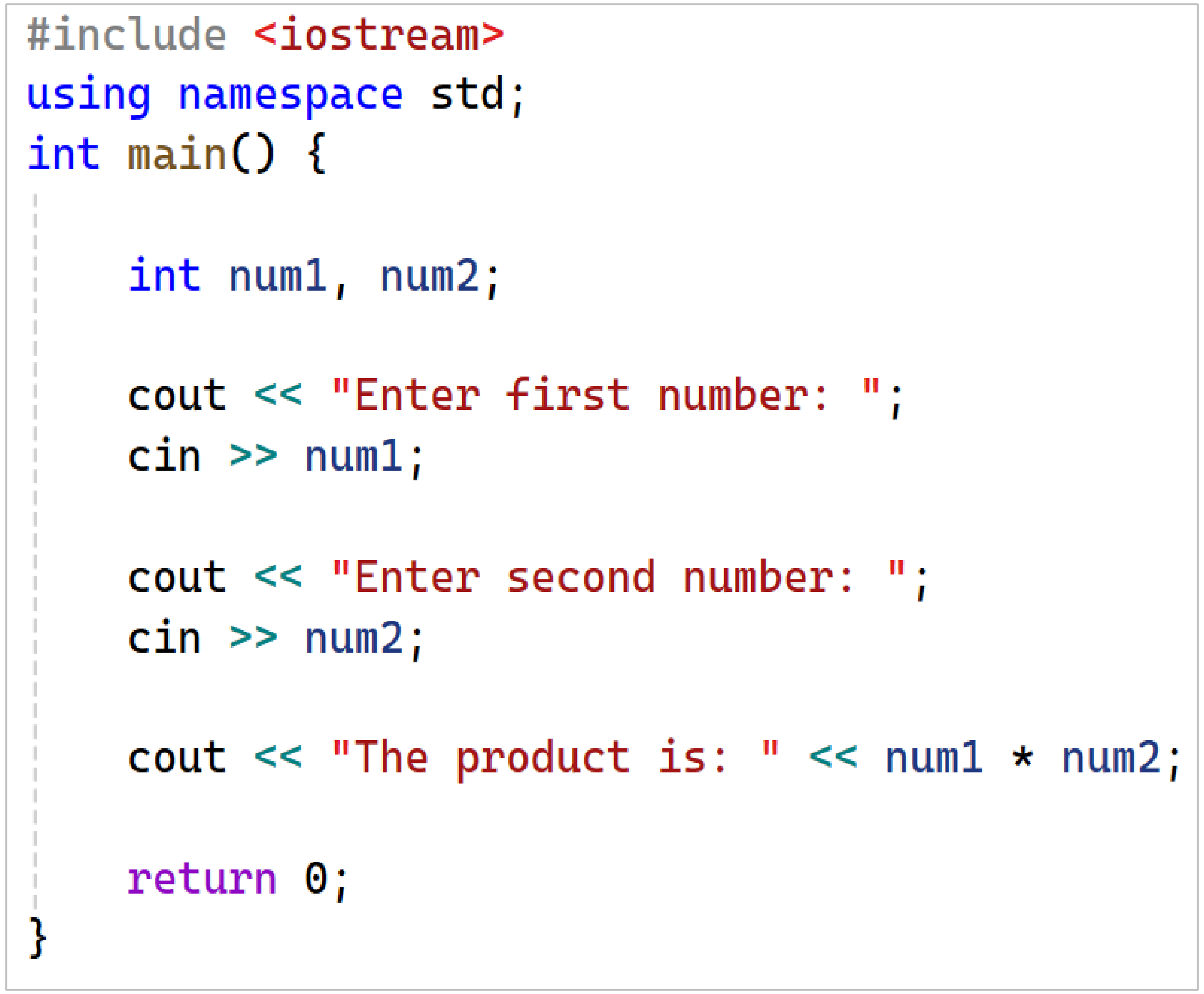
US-Letter (8,5" x 11")

[✓ Portrait] Landscape

Edit Data...

Clear Default Style

Flowchart (Homework)



```
#include <iostream>
using namespace std;
int main() {

    int num1, num2;

    cout << "Enter first number: ";
    cin >> num1;

    cout << "Enter second number: ";
    cin >> num2;

    cout << "The product is: " << num1 * num2;

    return 0;
}
```

Flowchart Steps:

1. Start
2. Input → num1
3. Input → num2
4. Process/Output → Display num1 * num2
5. End

Activities and Next Lecture's Topic

Activities

- Review this lecture note
- Practice

Next Lecture's Topic

- Selection Control Structures

References

- Gaddis, T. (2014). Starting out with C++: Early objects (7th ed.). Pearson Education.

Thank You!