CS-1002 Programming fundamentals (CYSEC)

Instructor : Jawad Hassan

jawad.hassan@nu.edu.pk

Fall 2022

September 13, 2022

FAST National University of Computer and Emerging Sciences, Islamabad

In Last week Lectures we have covered

- Data types
- Literals and variables
- Constants
- Arithmetic operators and Expressions

Today

- Type Conversion
- Coercion
- Type casting
- Output manipulation
- Working with Characters and string Objects
- Mathematical library function
- See Reading assignment at the end

Remember

- **Integer Division Vs Float Division**
- **Over flow and Under flow**

Type Conversion

1. *If an int is multiplied by a float, what data type will the result be?*
2. *What if a double is divided by an unsigned int?*

- Answer: C++ follows data type ranking:
    1. long double
    2. double
    3. float
    4. unsigned long
    5. long
    6. unsigned int
    7. Int
    8. Unsigned short
    9. Short
    10. char

Coercion

- When C++ is working with an operator, it strives to convert the operands to the same type. This automatic conversion is known as *type **coercion**.*
- When a value is converted to a higher data type, it is said to be ***promoted**.*
- To ***demote** a value means to convert it to a* lower data type.

Rules

1. chars, shorts, and unsigned shorts are automatically promoted to int.
2. When an operator works with two values of different data types, the lower ranking value is promoted to the type of the higher-ranking value.
3. When the final value of an expression is assigned to a variable, it will be converted to the data type of that variable.

Type Casting

- **To perform manual data type conversion.**

static_cast<DataType>(Value)

Output Manipulation/Formatting

Stream Manipulator	Description
`setw(` *n*`)`	Establishes a print field of *n* spaces.
`fixed`	Displays floating-point numbers in fixed point notation.
`showpoint`	Causes a decimal point and trailing zeroes to be displayed, even if there is no fractional part.
`setprecision(` *n*`)`	Sets the precision of floating-point numbers.
`left`	Causes subsequent output to be left justified.
`right`	Causes subsequent output to be right justified.

Mathematical Library Functions

Function	Example	Description
`abs`	`y = abs( x );`	Returns the absolute value of the argument. The argument and the return value are integers.
`cos`	`y = cos( x );`	Returns the cosine of the argument. The argument should be an angle expressed in radians. The return type and the argument are `doubles`.
`exp`	`y = exp( x );`	Computes the exponential function of the argument, which is `x`. The return type and the argument are `doubles`.
`fmod`	`y = fmod( x, z );`	Returns, as a `double`, the remainder of the first argument divided by the second argument. Works like the modulus operator, but the arguments are `doubles`. (The modulus operator only works with integers.) Take care not to pass zero as the second argument. Doing so would cause division by zero.
`log`	`y = log( x );`	Returns the natural logarithm of the argument. The return type and the argument are `doubles`.
`log10`	`y = log10( x );`	Returns the base-10 logarithm of the argument. The return type and the argument are `doubles`.
`sin`	`y = sin( x );`	Returns the sine of the argument. The argument should be an angle expressed in radians. The return type and the argument are `doubles`.
`sqrt`	`y = sqrt( x );`	Returns the square root of the argument. The return type and argument are `doubles`.
`tan`	`y = tan( x );`	Returns the tangent of the argument. The argument should be an angle expressed in radians. The return type and the argument are `doubles`.

Questions (???)

Thanks ☺

Reading Assignment

1. Formatting Output
   - ❑ Chapter 3 Topic 3.7
   - ❑ Page 138 to 148
2. Math Library functions
   - ❑ Chapter 3 Topic 3.9
   - ❑ Page 154 to 155

Book: STARTING OUT WITH **C++** **{BOOK-1-CPP}**

Author: Tony Gaddis

References

- **Book:** STARTING OUT WITH **C++**

  {BOOK-1-CPP}
- **Author: Tony Gaddis**

Working with Characters and string Objects

1. getline(cin, inputLine);
2. cin.get(ch);//ch = cin.get();
3. Cin.ignore();