Chapter 2

Data Types and Arithmetic Expressions

A Guide to this Instructor's Manual:

We have designed this Instructor's Manual to supplement and enhance your teaching experience through classroom activities and a cohesive chapter summary.

In addition to this Instructor's Manual, our Instructor's Resources also contain PowerPoint Presentations, Test Banks, and other supplements to aid in your teaching experience.

Overview

Chapter 2 covers variables and constants, data types, and arithmetic expressions.

Chapter Objectives

- Work with variables and constants
- Use the NSLog method to output data
- Use the Scanf method to read user input
- Learn about basic and derived data types
- Learn how to define a new data type with enumerations
- Use the preprocessor to create custom statements
- Create arithmetic expressions

Lecture Notes

- Although NSLog is an error log mechanism, we chose to use it here due to its ability to support the format specifiers that cannot be supported by `printf()`—an alternate mechanism for outputting data.
- `Scanf` is a tricky concept and does not work well with strings. When using %c to print a char, you may want to use %s instead as the %c will take the first char in the buffer and if the program is running for a second time there may be some garbage in the buffer from the previous run.
- In the enumerated data type, if a value is not assigned then the compiler will assign a value starting at 0 for the first enumerated type.

- Modulus is a tricky function for students to understand. Provide multiple examples for them to understand that it is the remainder.

Short Quiz:

Q1: How can you input and output data?
Answer: Use scanf to input and NSLog to output

Q2: What are the main types of data?
Answer: Basic and derived. Derived data types are a combination of basic data types.

Q3: What is an arithmetic expression?
Answer: It is a combination of operands and operator. Operands are numbers and operators are +, -, *,/ etc.