Input, Output, Variables and Expressions

CS 133N/ CS 161N

Mari Good

Objectives

- Introduce you to the steps involved in the programming process
- Introduce you to the Input Processing Output chart and its use in the program design process
- Introduce you to C# syntax for
  - Getting input from the user
  - Converting string input into other data types
  - Declaring and assigning values to variables
  - Creating expressions with arithmetic operators
  - Displaying formatted output to the user
- Practice with several examples

Programming is a Process

- Programming is a “problem solving activity”. Even experienced programmers need a set of tools to help them approach the solution of a programming problem.
  - Understand the problem
  - Outline a general solution using an IPO chart
  - Develop an algorithm for solving the problem using pseudocode
  - Test the algorithm for correctness
  - Translate the algorithm into C# syntax
  - Test and debug the C# program

Input Processing Output (IPO) Chart

- Is a relatively intuitive tool that helps a programmer develop a very high level solution to a problem by listing the input, output and processing steps required to transform the input into the output

IPO Chart

- Design and implement a program that asks the user to enter the price of a meal and the percent tip, calculates and displays the price, the amount of the tip and the total for the meal and the tip.
- Example

  15 price * 20 tipPercent * .01 = 3 tipAmount

  15 + 3 = 18 total

IPO Chart

- Input
  - price
  - tipPercent (20 means 20%)
- Processing
  - get the price
  - get the tipPercent
  - calculate the tipAmount
  - calculate the total
  - display the price, tipAmount and total
- Output
  - price
  - tipAmount
  - total

IPO Chart

- Design and implement a program that asks the user to enter his/her/their height in inches and weight in pounds, calculates and displays the user's Body Mass Index (BMI).

  BMI = (weight * 703) / (height * height)
- Example

  120 weight * 703 / 65 height * 65 = 19.97 bmi

IPO Chart

Input
- height in inches – whole number
- weight in pounds – whole number

Processing
- get the height
- get the weight
- calculate the bmi
- display the bmi

Output
- bmi

IPO Chart

- Design and implement a program that asks the user to enter a temperature in degrees fahrenheit, calculates and displays the same temperature in degrees celsius.

  c = 5/9 * (f - 32)
- Example

  32 degrees F is 0 degrees C

  5/9 * (32 – 32) = 0

IPO Chart

Input

Processing

Output

IPO Chart

- Design and implement a program that can be used with elementary school children to teach about change. The program should ask the student to enter a price that is less than 1 dollar. The program will calculate and display the amount of change due as well as the number of quarters, dimes, nickels and pennies.
- Example

  price = 34

  change = 100 – 34 = 66

  quarters = 66/25 = 2

  change = 66 – 2 * 25 = 16

IPO Chart

Input

Processing

Output

Questions?

- The temperature and the change problem that you just did are the first 2 problems of lab 2. The description of lab 2 in moodle contains 4 more (a total of 6) problems. In small groups
  - make sure you understand the problem
  - do an example
  - create the ipo chart for the last 4 problems.
- In a minute I'll show you how to convert all of the examples we've done so far into C# code.

Your first C# program

```
class Program
  {
      public static void Main()
      {
          Console.Write("Please enter your name: ");
          string name = Console.ReadLine();

          Console.WriteLine("Hello " + name);

          Console.ReadLine();
      }
  }
```

Main is the entry point for every C# program

Console is an object that represents the console window. You can use it to

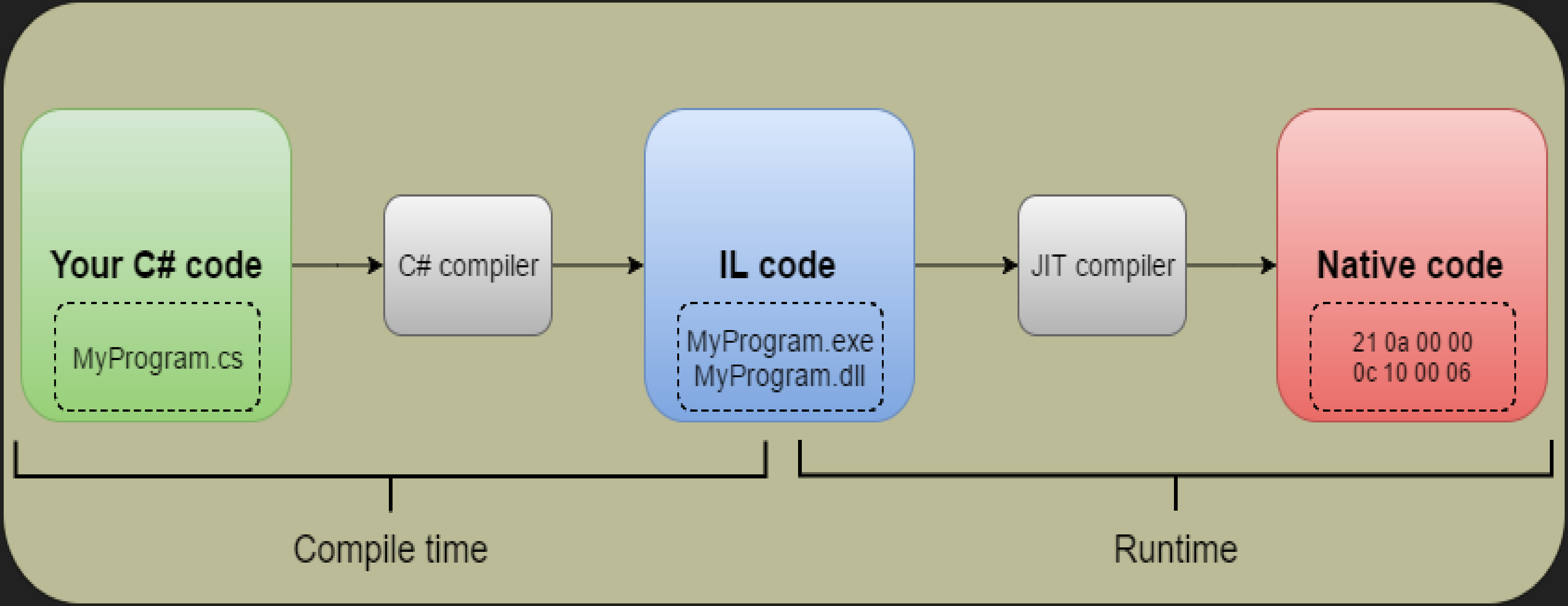
- Write and WriteLine are methods that write to the screen
- ReadLine is a method that reads a string from the keyboard

name is a variable

+ is the concatenation operator

" delimit strings

Compiling and Executing C# Code



IPO Chart

- Design and implement a program that asks the user to enter the price of a meal and the percent tip, calculates and displays the price, the amount of the tip and the total for the meal and the tip.
- Example

  15 price * 20 tipPercent * .01 = 3 tipAmount

  15 + 3 = 18 total

IPO Chart

Processing

- get the price
- get the tipPercent
- calculate the tipAmount
- calculate the total
- display the price, tipAmount and total

price, tipPercent, tipAmount and total are variables.
A variable is a named identifier used for storing data while a program is executing.

Translate the processing steps into C#

C# is a strongly typed language. All variables have a data type and must be declared. decimal is used for money. int for whole numbers.

```
public static void Main()
{
        Console.Write("Price: ");
        decimal price = decimal.Parse(Console.ReadLine());
        Console.Write("Tip Percentage (Enter 20 for 20%): ");
        int tipPercent = int.Parse(Console.ReadLine());
```

Keyboard input is a set of characters. To store it in a variable that represents a number you have to parse it. Each data type has its own parse method

Translate the processing steps into C#

Notice that every variable must be declared before it can be used. Variable names in C# should be camelCase.

= is the assignment operator. It is used to give a variable a value. Notice the variable is on the left hand side.

```
decimal tipAmount = price * tipPercent * .01M;
decimal total = price + tipAmount;
```

Even literal values like .01 have a data type in C#. Real literals are double by default. M makes .01 a decimal.

* and + are arithmetic operators. All operators have an "order of operation" or precedence. Multiplication and division are executed before addition and subtraction, from left to right. ( ) change that default order.

Translate the processing steps into C#

ToString is a method that converts almost any data type to a string or set of characters. The “C” is a format specification that makes the value look like money.

```
        Console.WriteLine("Price: " + price.ToString("c"));
        Console.WriteLine("Tip Amount: " + tipAmount.ToString("c"));
        Console.WriteLine("Total: " + total.ToString("c"));
}
```

+ is the concatenation operator

More examples

- Let's look at this program and a couple other examples in dotnetfiddle.net
- Then we'll use dotnetfiddle.net to do the first 2 problems from the lab together
- Finally, there will be time to do the other 4 problems from the lab (you've already done the IPO charts) in small groups. I'll help whenever you get stuck.

What's Next

- Developing algorithms in pseudocode
- Syntax
  - if statement
  - Relational operators
  - Logical operators
  - switch statement
- Don't forget
  - Reading Quiz 2
  - Programming Quiz 2
  - Lab 2 – 6 problems