

# Algorithms

---

- Every program is based on an algorithm, or more than one algorithm
- An algorithm is like a recipe for cooking
  - It tells the ingredients (inputs)
  - It tells the sequential steps for processing the inputs
  - It tells the serving size and style (outputs)
- The computer acts like a chef, exactly following the algorithm recipe

# The Software Development Process

---

Understand  
the problem

Develop and  
describe an  
algorithm

Test the algorithm  
with simple inputs

Translate  
the algorithm  
into C++

Compile and test  
your program

For each problem  
the programmer goes  
through these steps

*You MUST write an algorithm  
in words, pictures, and/or  
equations before attempting  
to translate to C++*

# Describing an Algorithm with Pseudocode

---

## Pseudocode

- An informal description
- Not in a language that a computer can understand, but easily translated into a high-level language (like C++).

# Three Important Properties of an Algorithm

The **algorithm** described in pseudocode must be

- Unambiguous
  - There are precise instructions for what to do at each step
  - and where to go next.
- Executable
  - Each step can be carried out in practice.
- Terminating
  - It will eventually come to an end.

# Example of Pseudocode

---

Consider this problem:

- You have the choice of buying two cars.
- One is more fuel efficient than the other, but also more expensive.
- You know the price and fuel efficiency (in miles per gallon, mpg) of both
- You plan to keep the car for ten years.
- Assume a price of gas is \$4 per gallon and usage of 15,000 miles per year.
- You will pay cash for the car (no financing costs)

Which car is the better deal?

# Algorithm Pseudocode, Step 1: Determine I/O

## Step 1 Determine the inputs and outputs.

In our sample problem, we have these inputs:

- *purchase price1* and *fuel efficiency1*  
the price and fuel efficiency (in mpg) of the first car
- *purchase price2* and *fuel efficiency2*  
the price and fuel efficiency of the second car

We simply want to know which car is the better buy.  
That is the desired output.

# Algorithm Pseudocode, Step 2: Decompose

**Step 2** Break down the problem into smaller tasks.

What will we do for ***each*** car?

1. The total cost for a car is  
*purchase price + operating cost*
2. We assume a constant usage and gas price for ten years, so the operating cost depends on the cost of driving the car for one year.  
The operating cost is  
*10 x annual fuel cost*
3. The annual fuel cost is  
*price per gallon x annual fuel consumed*
4. The annual fuel consumed is  
*annual miles driven / fuel efficiency*

## Algorithm Pseudocode, Step 3

**Step 3** Describe each subtask in pseudocode.

You will need to arrange the steps so that any intermediate values are computed before they are needed in other computations.

For each car, compute the total cost as follows:

```
annual fuel consumed = annual miles driven / fuel efficiency
annual fuel cost = price per gallon x annual fuel consumed
operating cost = 10 x annual fuel cost
total cost = purchase price + operating cost
```

```
If total cost1 < total cost2
```

```
    Choose car1
```

```
Else
```

```
    Choose car2
```

# Algorithm Pseudocode, Step 4: Testing

**Step 4** Test your pseudocode by working a problem.

Use these sample values:

- Car 1: \$25,000, 50 miles/gallon
- Car 2: \$20,000, 30 miles/gallon

FIRST CAR:

**annual fuel consumed = 1500 / 50 = 300**

**annual fuel cost = 4 × 300 = 1200**

**operating cost = 10 × 1200 = 12000**

**total cost = 25000 + 12000 = 37000**

SECOND CAR:

*(let's assume you can do the math)* **total cost = 40000**

**If total cost1 < total cost2 ...**

The algorithm says: **choose the FIRST CAR**

## Topic 8

---

1. What is programming?
2. Anatomy of a computer
3. Machine code and programming
4. Becoming familiar with your programming environment
5. Analyzing your first program
6. Errors
7. Problem solving: algorithm design
8. Chapter summary

# Chapter Summary

---

## **Describe the building blocks of a simple program.**

- Every C++ program contains a function called `main`.
- `cout` and the `<<` operator to display values on the screen.
- Enclose text strings in quotation marks.
- Use `+` to add two numbers and `*` to multiply two numbers.
- Send `endl` to `cout` to end a line of displayed output.
- End each statement with a semicolon.

## **Write pseudocode for simple algorithms.**

- Pseudocode is an informal description of a sequence of steps for solving a problem.
- An algorithm for solving a problem is a sequence of steps that is unambiguous, executable, and terminating.

## **Classify program errors as compile-time and run-time errors.**

- A compile-time error is a violation of the programming language rules that is detected by the compiler.
- A run-time error causes a program to take an action that the programmer did not intend.
- The programmer is responsible for inspecting and testing the program to guard against run-time errors.