Algorithms

- **algorithm**: A list of steps for solving a problem.

An algorithm for baking sugar cookies:
- Mix the dry ingredients.
- Cream the butter and sugar.
- Beat in the eggs.
- Stir in the dry ingredients.
- Set the oven for the appropriate temperature.
- Set the timer.
- Place the cookies in the oven.
- Take cookies out when ready.
- Mix the ingredients for the frosting.
- Spread frosting and sprinkles onto the cookies.

A structured solution

1. Make the cookie batter.
   - Mix the dry ingredients.
   - Cream the butter and sugar.
   - Beat in the eggs.
   - Stir in the dry ingredients.
2. Bake the cookies.
   - Set the oven for the appropriate temperature.
   - Set the timer.
   - Place the cookies into the oven.
   - Allow the cookies to bake.
3. Add frosting and sprinkles.
   - Mix the ingredients for the frosting.
   - Spread frosting and sprinkles onto the cookies.

Redundancy

- How would we bake a double batch of cookies?

**Unstructured:**
- Mix the dry ingredients.
- Cream the butter and sugar.
- Beat in the eggs.
- Stir in the dry ingredients.
- Set the oven for the appropriate temperature.
- Set the timer.
- Place the first batch of cookies into the oven.
- Allow the cookies to bake.
- Set the oven ...  
- Set the timer.
- Place the second batch of cookies into the oven.
- Allow the cookies to bake.

**Structured:**
1. Make the cookie batter.
2a. Bake the first batch of cookies.
2b. Bake the second batch of cookies.
3. Add frosting and sprinkles.

Observation: Higher-level operations help eliminate redundancy.

Drawing a rectangle

Suppose we want to write a program that draws an ASCII rectangle such as:

```
************
*          *
*          *
*          *
************
```

**Unstructured solution**

```
print "************"
print "*          *"
print "*          *"
print "*          *"
print "************"
```

**redundancy**: Occurrence of the same sequence of commands multiple times in a program
Drawing a rectangle revisited

```python
def drawRectangle():
    drawLine()
    drawSides()
    drawLine()

def drawLine():
    print "***********"

def drawSides():
    print "*              *
    print "*              *
    print "*              *
    print "*              *"
drawRectangle()
```

Functions

- **function**: A group of statements which is given a name.
  - **procedural decomposition**: breaking a problem into subproblems

  - using functions has two steps:
    1. **define** it (write down the recipe)
       - write a group of statements and give it a name
    2. **call** it (cook using the recipe)
       - tell the program to execute the method

  - useful for:
    - breaking a large program into smaller pieces
      - Makes it easier to write, and test
      - Easier to understand afterwards
      - Can divide the task
    - eliminating redundancy through reuse

Program flow

- When a method is called:
  - the execution "jumps" into that method,
  - executes all of its statements, and then
  - "jumps" back to the statement after the method call.

  - Trace the execution of the rectangle program

Functions that take parameters

- Let’s rewrite the celsius to fahrenheit program using functions