# 4.1 - Algorithms

# Algorithms

*Algorithm*: A finite set of unambiguous instructions that, given some set of initial conditions, can be performed in a prescribed sequence to achieve a certain goal and that has a recognizable set of end conditions.

In general, a good algorithm has three properties:

1. *Compact*: The length of the algorithm should be as short as possible, without sacrificing efficiency and ease of use
2. *General*: The algorithm will work with different sizes of tasks or paremeters
3. *Abstract*: The algorithm does not depend on a particular programing language or computer system

# Approximating Pi

The Madhava-Leibniz series uses the following to approximate $\pi$:

$$\pi = \sqrt{12} \sum_{k=1}^{\infty} \frac{(-\frac{1}{3})^k}{2k+1}$$

$\pi$ was known to 7 digits in Chinese mathematics and 5 to Indian mathematics in the 5th century. It wasn't until the 14th century that this series was discovered, increasing the number of known digits of pi to 11. We now know pi to 50 trillion digits.

# Approximating Pi

$$\pi = \sqrt{12} \sum_{k=1}^{\infty} \frac{(-\frac{1}{3})^k}{2k+1}$$

This series represents an algorithm. It qualifies for the three main principles of an algorithm:

1. Efficient
2. General
3. Abstract

# Algorithms

To help describe and create algorithms, we rely on flowcharts and psuedocode.

*Flowcharts* are a visual representation of how you can "move" through the algorithm. There are shapes that represent the various steps and calculations needed.

*Psuedocode* is fake code. A better description might be syntax that represents code, but isn't syntatically correct.

# Flowchart Example