Methods

Main Reason: Help people solve programming problems by divide and conquer.

Programming Problem: Given: Some idea or specification for what a computer program or part of one should do.

Solution: An actual, written program or part that does it!

A program is like a script in a play. But—separately from the rest, it can have loops and conditions and it can throw loops and conditions and it can be divided out and conquered one by one.

void displayMsg(int)
{
    system.out.println(""Msg");
}

void Dispaly(int times)
{
    int i = 0;
    while (i < times)
    {
        System.out.println(""Msg");
        i = i + 1;
    }
}

The initial value a parameter is given by the caller.
① a method can divide and a calculation that results in a value:

\[
\text{finalVelocity}(\text{initVel, acc, time}) = \text{initVel} + \text{acc} \times \text{time}
\]

```java
def finalVelocity(double initVel, double acc, double time):
    return initVel + acc * time;
```

③ method can have parameter n arguments
They customize the data that the code or calculation depends on
Methods can have local variables, like i in display. A parameter (like times) is a kind of local variable.

```java
void display (int times) { int i = 0;
  while (i < times)
  { System.out.println ("msg");
    ++i;
  }
}
```

Local variables in one method have nothing to do with local variables in other methods, even if they have the same name, like i.

```java
void display (int times) {
  if (times == 0) return;
  else
  { System.out.println (" ");
    display (time - 1);
  }
}
```

C / C++

`exit (0)`

Makes the whole program just die.

```java
void display (int times) {
  if (times != 0) {
    System.out.println (" ");
    display (times - 1);
  }
}
```
1. Declare "times" to be an int variable.
2. Its initial value comes from the caller.

```cpp
void display (int times)
{
    while (times > 0)
    {
        system.out.println("Msg");
        times--;
    }
}
```