Methods

INFS1609/COMP1400 – Week 2

Methods are like functions

- A method is like a function in mathematics
- It maps parameters to the function value

Methods are like functions

- A method is like a function in mathematics
- It maps parameters to the function value

```
int multiply(int x, int y)
{
     return x * y;
}
```

Methods are like functions

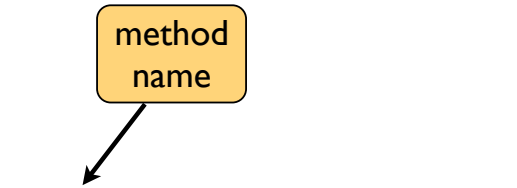
- A method is like a function in mathematics
- It maps parameters to the function value

```
int multiply(int x, int y)
{
     return x * y;
}
```

```
multiply(2, 3);
6 (int)
```

What are methods for?

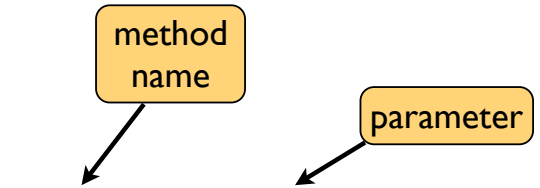
- We use methods to define operations that we will use more than once
- They save effort and make the program more readable
- They allow code re-use.



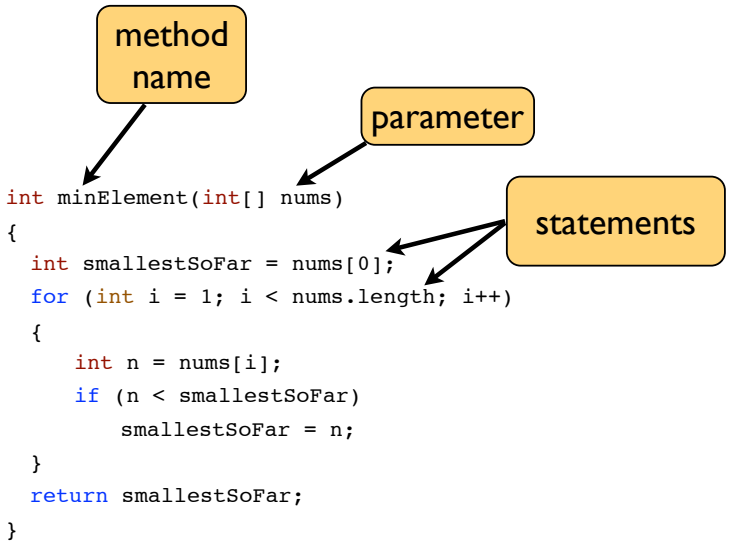

```
int minElement(int[] nums)
{
  int smallestSoFar = nums[0];
  for (int i = 1; i < nums.length; i++)
  {
      int n = nums[i];
      if (n < smallestSoFar)
          smallestSoFar = n;
  }
  return smallestSoFar;
}
```

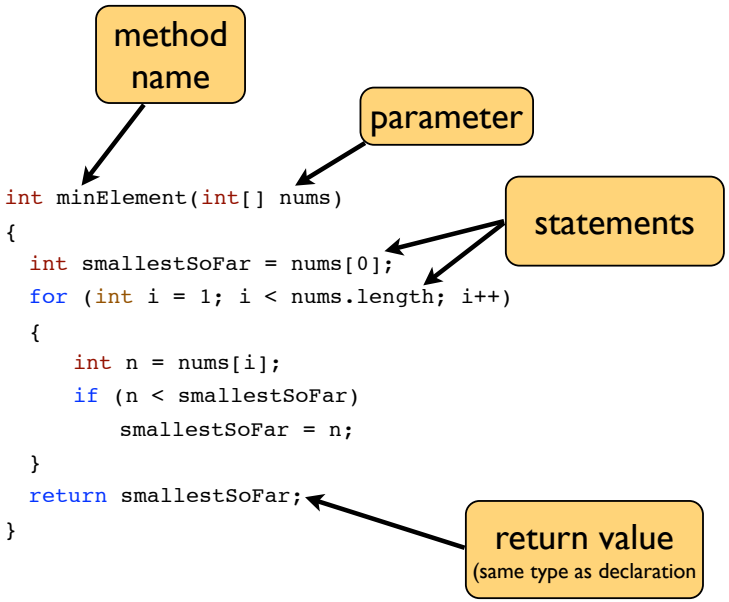
Method Components

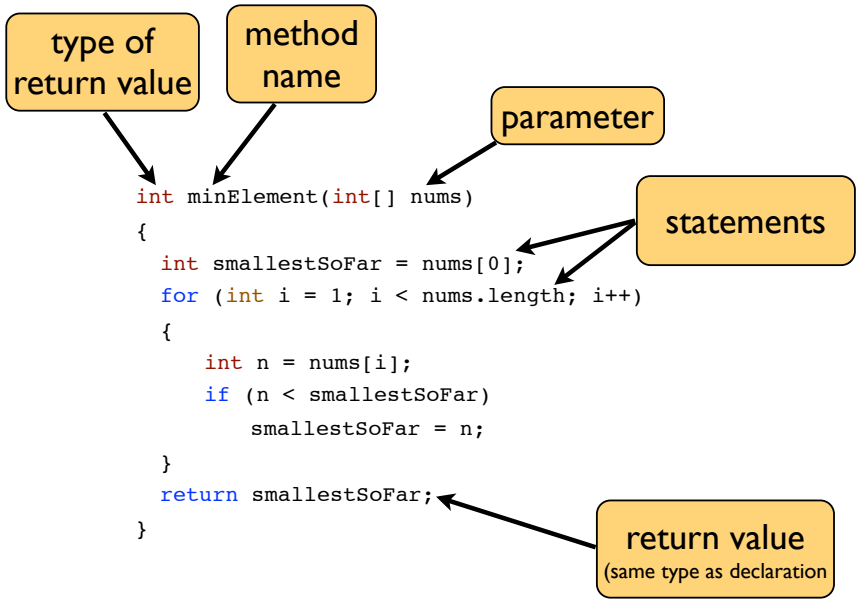
```
int minElement(int[] nums)
{
  int smallestSoFar = nums[0];
  for (int i = 1; i < nums.length; i++)
  {
      int n = nums[i];
      if (n < smallestSoFar)
          smallestSoFar = n;
  }
  return smallestSoFar;
}
```

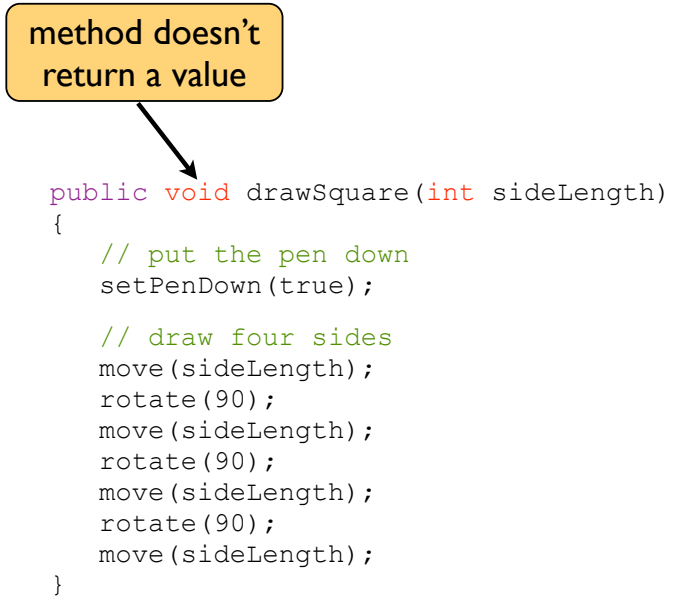



```
int minElement(int[] nums)
{
  int smallestSoFar = nums[0];
  for (int i = 1; i < nums.length; i++)
  {
      int n = nums[i];
      if (n < smallestSoFar)
          smallestSoFar = n;
  }
  return smallestSoFar;
}
```


method name
parameter
statements
int minElement(int[] nums)
{
int smallestSoFar = nums[0];
for (int i = 1; i < nums.length; i++)
{
int n = nums[i];
if (n < smallestSoFar)
smallestSoFar = n;
}
return smallestSoFar;
}


method name
parameter
statements
int minElement(int[] nums)
{
int smallestSoFar = nums[0];
for (int i = 1; i < nums.length; i++)
{
int n = nums[i];
if (n < smallestSoFar)
smallestSoFar = n;
}
return smallestSoFar;
}
return value
(same type as declaration


type of return value
method name
parameter
statements
int minElement(int[] nums)
{
int smallestSoFar = nums[0];
for (int i = 1; i < nums.length; i++)
{
int n = nums[i];
if (n < smallestSoFar)
smallestSoFar = n;
}
return smallestSoFar;
}
return value
(same type as declaration

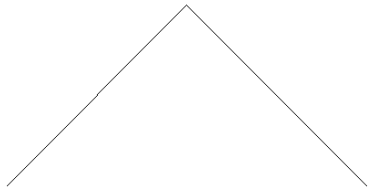

method doesn't return a value
public void drawSquare(int sideLength)
{
// put the pen down
setPenDown(true);
// draw four sides
move(sideLength);
rotate(90);
move(sideLength);
rotate(90);
move(sideLength);
rotate(90);
move(sideLength);
}

Methods don't have to have parameters

```
void dragonCurve()
{
    dragonSegment(12, rightTurn);
}
```

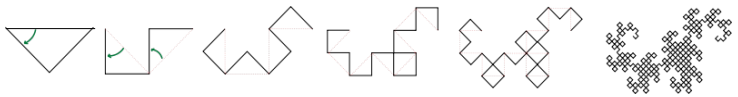
```
dragonCurve();
```

The Dragon Curve



The Dragon Curve

- Replace a line segment with a right angle
- For each new segment repeat replacing each line with a right angle, alternating between left turn and right turn



Don't Panic!

Recursion

```
final int rightTurn = 90;
final int leftTurn = -90;
final int length = 5;

public void dragonSegment(int depth, int turn)
{
    if (depth == 0)
        move(length);
    else
    {
        depth--;
        dragonSegment(depth, rightTurn);
        rotate(turn);
        dragonSegment(depth, leftTurn);
    }
}
```