

ESSENTIAL ARRAY ALGORITHMS

Sum / Average

Strategy:

- Use a variable to keep track of a running total.
- Declare the variable outside the loop, set to 0.
- Update the total within the loop.
- Return the total (or total / arrayName.length for average).

```
static double averageHeight(double[] heights)
{
    double total = 0;

    for (double h : heights)
    {
        total += h;
    }
    return total / heights.length;
}
```

Finding Maximum / Minimum

Strategy:

- Use a variable to keep track of the current max value.
- Declare the variable outside the loop, set to arrayName[0].
- Overwrite the current max value inside of an if statement.
- Return the max value.

```
static int getHighScore(int[] scores)
{
    int currentHighScore = scores[0];

    for (int score : scores)
    {
        if (score > currentHighScore)
        {
            currentHighScore = score;
        }
    }
    return currentHighScore;
}
```

Counting Matches

Strategy:

- Use a variable to keep track of a count.
- Declare the variable outside the loop, set to 0.
- Increment the count inside an if statement within the loop
- Return the count.

```
static int countAGrades(double[] grades)
{
    int count = 0;

    for (double grade : grades)
    {
        if (grade >= 90)
        {
            count++;
        }
    }
    return count;
}
```

Checking for Specific Value

Strategy:

- Let the method take an array and a search value as input.
- Return true within the loop inside of an if statement.
- Return false outside of the loop.

```
static boolean contains(double[] values, double
searchVal)
{
    for (double v : values)
    {
        if (v == searchVal)
        {
            return true;
        }
    }
    return false;
}
```

Modify Array Contents

Strategy:

- Use a count-controlled for loop to increment the index.
- Set each element to the new value in the loop body.
- Return the array (unless you are writing a void method).

```
static double[] applyFee(double[] accounts)
{
    for (int i = 0; i < accounts.length; i++)
    {
        accounts[i] = accounts[i] - 18.99;
    }
    return accounts;
}
```

Note: only the count-controlled for loop can modify the elements of an array. The collection-controlled "for each" loop can only read values from the array.