```
SomeObject[] someObj = new SomeObject[N];
SomeObject tempObj = new SomeObject();  //Create temporary
someObj[i+j] = tempObj;
tempObj.foo(k);                         //Use it...
tempObj.foo(k+1);
tempObj.foo(k+2);
tempObj.foo(k+3);
tempObj.foo(k+4);
```

For many iterations, the optimized code is twice as fast as the original code, and it is smaller.

Loop Unrolling

Unrolling a loop has the advantage of eliminating the code for the loop construct and branching, thereby resulting in faster execution. Its disadvantage is that more code is generated, thereby resulting in a larger `.class` file. A typical loop might be coded like this:

```
int[] ia = new int[4];
for (int i=0; i<4; i++)
  ia[i] = 10;
```

and unrolled like this:

```
int[] ia = new int[4];
ia[0] = 10;
ia[1] = 10;
ia[2] = 10;
ia[3] = 10;
```

This particular unrolled loop is about 7 percent faster than the original version.

Algebraic Simplification

Algebraic simplification involves using the rules of algebra to simplify expressions. This simplification can lead to smaller and faster code. For example, consider the following code:

```
int a = 1; int b = 2; int c = 3;
int d = 4; int e = 5; int f = 6;
int x = (f*a)+(f*b)+(f*c)+(f*d)+(f*e);
```