# Exercise 9 Answers

# Arrays

---

1. (a)

   (b) Add the following line to the end of the `doStuff` method:

   ```
   double[] nums = new double[10];
   ```

   (c)
   ```
   for (int i=0; i<10; i++) {
      nums[i] = 1 + i * 0.1;
   }
   ```

   (d)
   ```
   for (int i=0; i<10; i++) {
      System.out.println(nums[i]);
   }
   ```

   (e)
   ```
   ArrayPrint.printArray(nums);
   ```

   (f) Define the method like so:

   ```
   public void printArray(double[] x) {
      for (int i=0; i<x.length; i++) {
         System.out.println(x[i]);
      }
   }
   ```

   (g)
   ```
   printArray(nums);
   ```

2. (a)

   (b) Add the following line to the end of the `doStuff` method:

   ```
   Human[] threesome = new Human[3];
   ```

   (c)
   ```
   threesome[0] = new Human("Angus");
   threesome[1] = new Human("Brian");
   threesome[2] = new Human("Charles");
   ```

   As an alternative to the last four lines of code that define and initialise the `threesome` array, it is possible to define and initialise the `threesome` array using a special compound statement that will be familiar to C programmers:

   ```
   Human[] threesome = {new Human("Angus"),
                        new Human("Brian"),
                        new Human("Charles")};
   ```

   I will not use this technique in these examples because I consider it to be a little obscure.

   (d)
   ```
   for (int i=0; i<2; i++) {
      System.out.println(threesome[i]);
   }
   ```

   Alternatively, the following does the same thing:

   ```
   for (int i=0; i<2; i++) {
      System.out.println(threesome[i].toString());
   }
   ```

(e) Define the method like so:

```
public void printArray(Human[] x) {
   for (int i=0; i<x.length; i++) {
      System.out.println(x[i]);
   }
}
```

(f) `ArrayPrint.printArray(threesome);`