

## Getting started

Make sure you have followed the instructions on the tools handout relevant to directory setup and using Java™ before you begin development.

Readings for the whole problem set are Chapters 1 through 5 of the Java Tutorial, 3rd Edition. Readings for the individual problems are listed with each problem.

If you make any assumptions in solving any problem in the exercise, please ensure that they are clearly stated in your solutions. Assumptions must be coherent and reasonable.

## Problem 1: Language Basics

**Readings: Language Basics: Variables, Operators, Expressions, Statements, and Blocks**

Look at the code below which calculates the radius of a circle in feet and inches, given that the circle has a given area. Download source (JAVA).

```
public class MathCalc
{
    public static void main(String[] args)
    {
        double radius = 0.0;
        double circleArea = 100.0;
        int feet = 0;
        int inches = 0;
        radius = Math.sqrt(circleArea/Math.PI);
        feet = (int)Math.floor(radius);
        inches = (int)Math.round(12.0*(radius - feet));
        System.out.println("The radius of a circle with area" +
circleArea
                           + " square feet is\n " + feet + " feet " + inches + "
inches");
    }
}
```

The diameter of the Sun is approximately 865,000 miles. The diameter of the Earth is approximately 7600 miles. Use the methods in the class Math to calculate,

- (a) the volume of the Earth in cubic miles
- (b) the volume of the Sun in cubic miles
- (c) the ratio of the volume of the Sun to the volume of the Earth

and then output the three values. Treat both the earth and sun as spheres. The volume of a sphere is given by the formula  $\frac{4}{3} \pi r^3$  where  $r$  is the radius.

Run the code you have written, and turn in both your code and the output of the program. The output should say something like: The volume of the Earth is X cubic miles, the volume of the sun is Y cubic miles, and the ratio of the volume of the Sun to the volume of the Earth is Z.

## Problem 2: Language Basics

**Readings: Language Basics : Control Flow Statements**

```
public class Primes
{
    public static void main(String[] args)
    {
        int nValues = 50;
        boolean isPrime = true;
        for(int i = 2; i <= nValues; i++)
        {
            isPrime = true;
```

```

        for (int j = 2; j < i; j++)
        {
            if (i % j == 0)
            {
                isPrime = false;
                break;
            }
        }
        if (isPrime)
            System.out.println(i);
    }
}

```

Create a new program that has the following features. Download source (JAVA).

- (a) Uses **labeled continue** instead of break.
- (b) Does not require the isPrime variable.
- (c) When testing whether an integer is prime, it is sufficient to try and divide by integers up to the square root of the number being tested.

Turn in your rewritten code.

### Problem 3: Object Basics

**Readings:** Object Basics and Simple Data Objects, The Life Cycle of an Object, Characters and Strings

In the following code the **soliloquy** is analyzed character by character to determine the vowels, spaces and letters used. **Fill in the code that computes the number of spaces, vowels, and consonants**. Download source (JAVA).

```

public class StringCharacters
{
    public static void main(String[] args)
    {
        String text = "To be or not to be, that is the question;"
        +"Whether `tis nobler in the mind to suffer"
        +" the slings and arrows of outrageous fortune,"
        +" or to take arms against a sea of troubles,"
        +" and by opposing end them?";
        int spaces = 0,
        vowels = 0,
        letters = 0;
        //YOUR CODE HERE
        System.out.println("The text contained vowels: " + vowels + "\n"
+
        consonants " + (letters - vowels) + "\n"+ spaces: " + spaces);
    }
}

```

Run the program once you have written your code. Turn in your code, and the output produced when the code runs.

### Problem 4: Object Basics

**Readings:** Object Basics and Simple Data Objects

**Numbers**

**Arrays**

Write a program that sets up a String variable with the soliloquy in the previous question, extracts the words from the text and sorts them into alphabetical order. You may define 'words' however you wish (within reason), but provide your definition with your solution. You

can use the sorting method of your choice. The simplest one is the bubble sort, which works as follows:

- Starting with the first element in the array compare successive elements (0 and 1, 1 and 2, 2 and 3, and so on).
- If the first element of any pair is greater than the second, interchange the two elements.
- Repeat the process for the whole array until no interchanges are necessary. The array elements will now be in ascending order.
- Run the program once you have written your code. Turn in your code, and the output produced when the code runs.