COMP90041
Programming and Software Development
2020 - Semester 1
Lab 2 - Week 3

Yuhao(Beacon) Song

yuhsong1@unimelb.edu.au

Introduction

- Master of Information Technology (Artificial Intelligence), Second Year

- Timetable
  - Tue(11) 14:15-15:15 (Melbourne Time) https://unimelb.zoom.us/j/4631577274
  - Tue(07) 16:15-17.15 (Melbourne Time) https://unimelb.zoom.us/j/4631577274

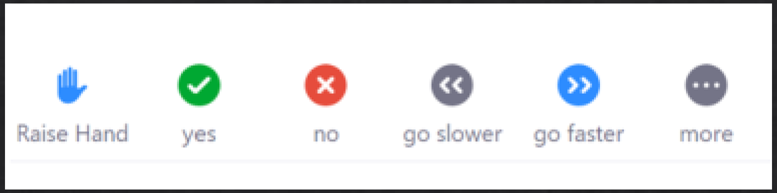
- Contact
  - yuhsong1@unimelb.edu.au
  - Github: https://github.com/Beaconsyh08/COMP90041_Programming_and_Software_Development_Tutorials.git

Outline

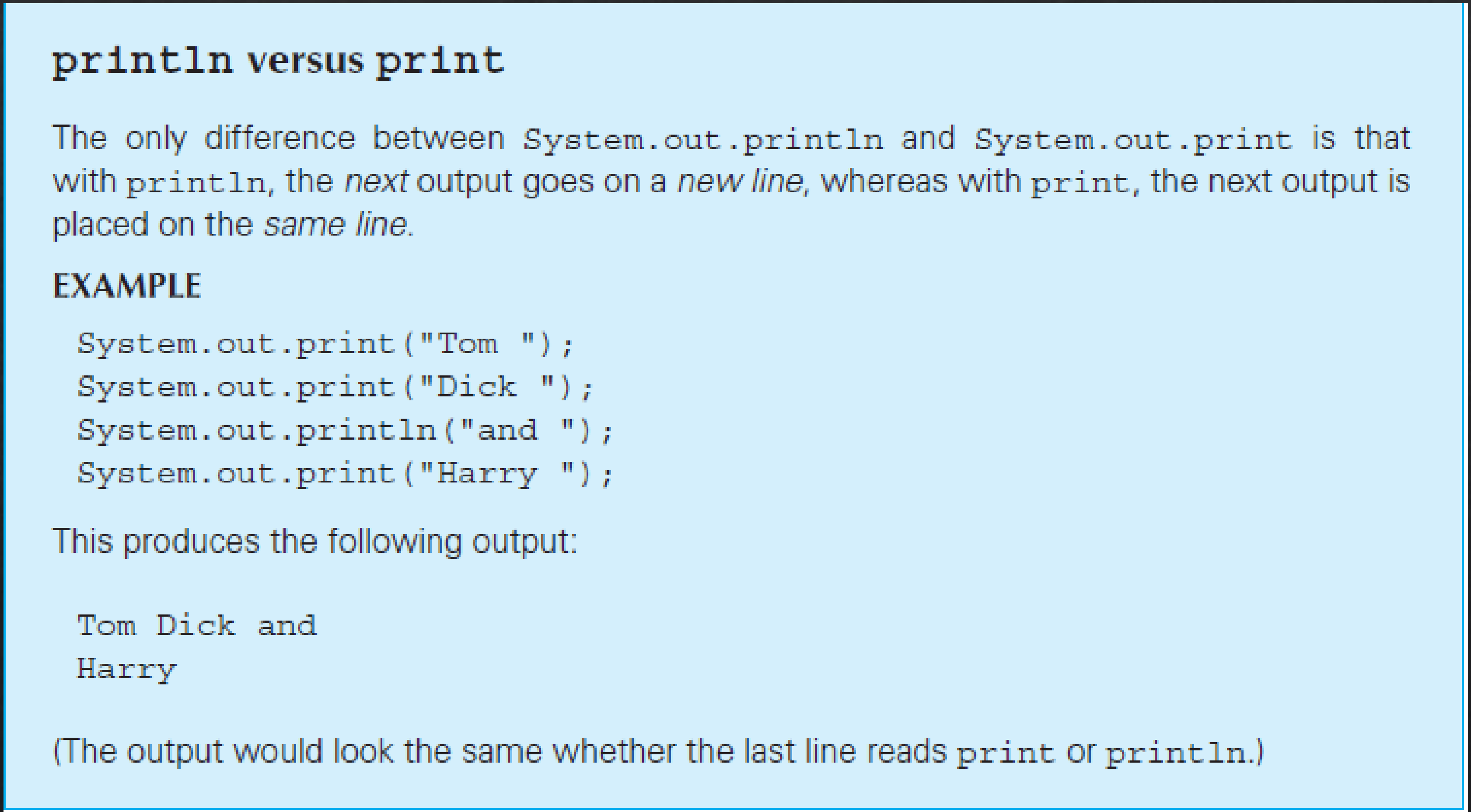
- Brief Introduction for Zoom
- Lab 1 Explanation(optional)
- Lecture Review
- Exercise & demo

Zoom

- Download
  - https://www.unimelb.edu.au/zoom/home
- Basic Function
  - Yes & No
  - Raise Hand
  - Poll
- Preparation
  - Food
  - Drink



Lecture Review – Print Function



`println` versus `print`

The only difference between `System.out.println` and `System.out.print` is that with `println`, the *next* output goes on a *new line*, whereas with `print`, the next output is placed on the *same line*.

EXAMPLE

```
System.out.print("Tom ");
System.out.print("Dick ");
System.out.println("and ");
System.out.print("Harry ");
```

This produces the following output:

```
Tom Dick and
Harry
```

(The output would look the same whether the last line reads `print` or `println`.)

Lecture Review – Scanner

Console Input Using the Scanner Class

- The following line creates an object of the class Scanner and names the object keyboard :

  Scanner keyboard = new Scanner(System.in);

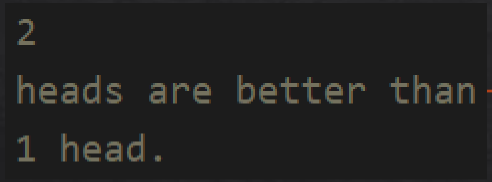
- Although a name like keyboard is often used, a Scanner object can be given any name
  - For example, the Scanner object is named scannerObject

    Scanner s1 = new Scanner(System.in);

    s1.nextLine/ nextInt/ Next

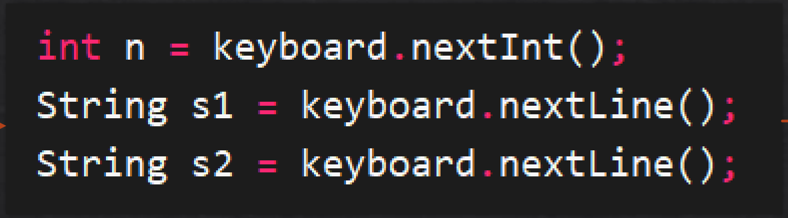
QUIZ

Input



```
2
heads are better than
1 head.
```

Code

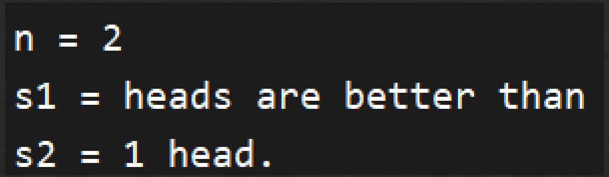


```java
int n = keyboard.nextInt();
String s1 = keyboard.nextLine();
String s2 = keyboard.nextLine();
```

Output

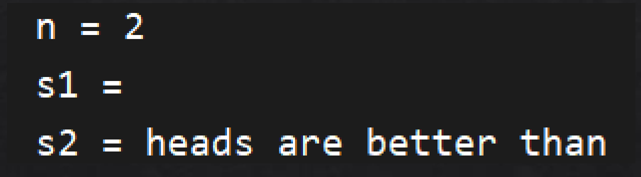
PITFALL: Dealing with the Line Terminator, '\n'

Expected



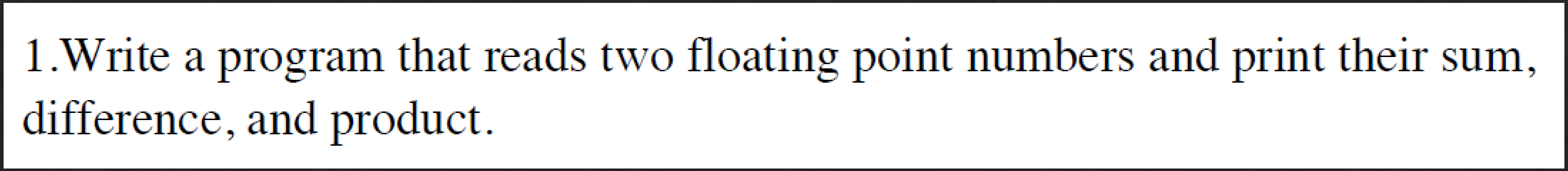
```
n = 2
s1 = heads are better than
s2 = 1 head.
```

Reality

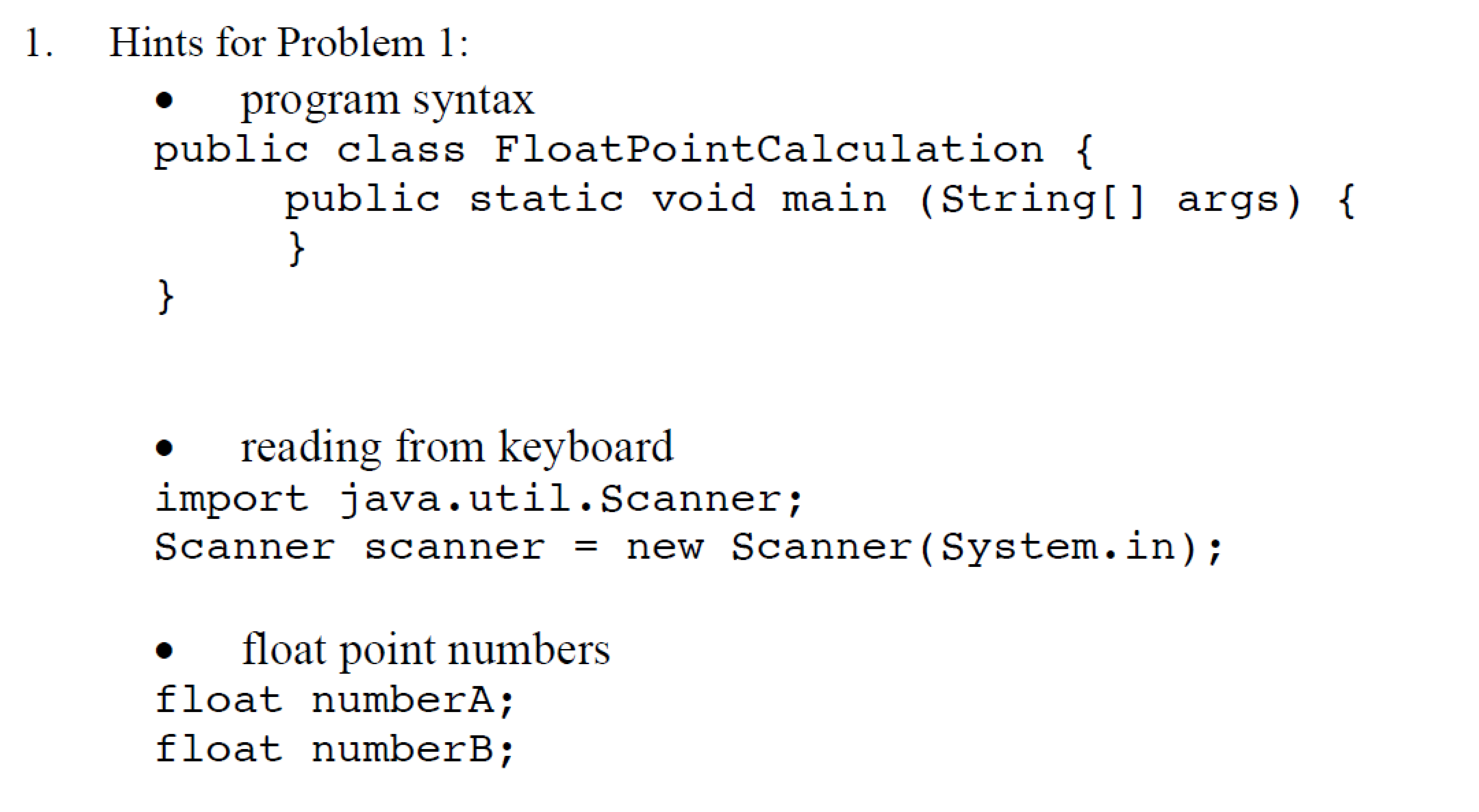


```
n = 2
s1 =
s2 = heads are better than
```

Tutorial Q1



1.Write a program that reads two floating point numbers and print their sum, difference, and product.



1. Hints for Problem 1:

- program syntax

```
public class FloatPointCalculation {
    public static void main (String[] args) {
    }
}
```

- reading from keyboard

```
import java.util.Scanner;
Scanner scanner = new Scanner(System.in);
```

- float point numbers

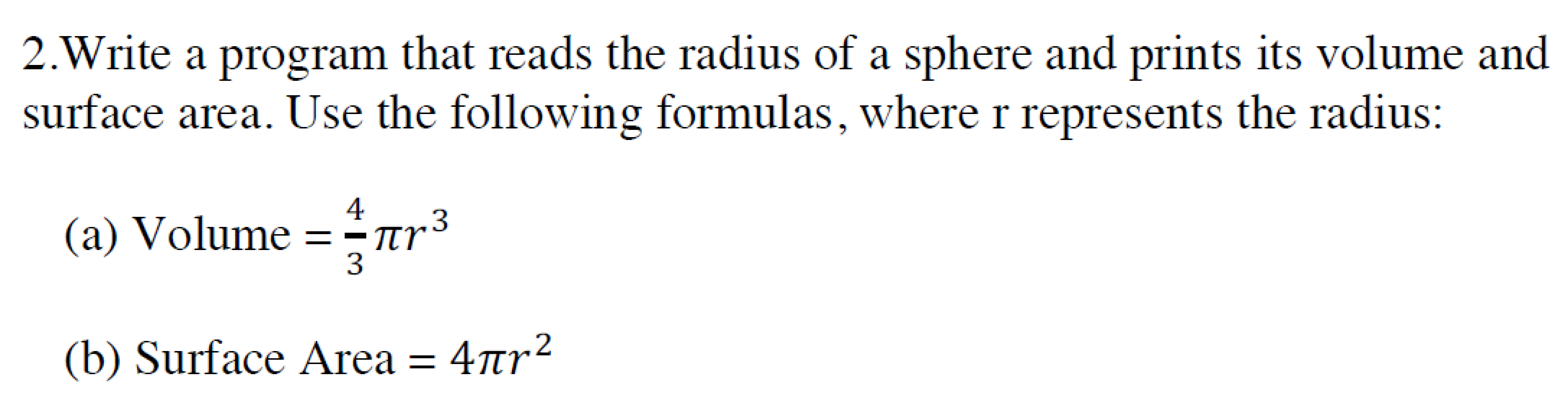
```
float numberA;
float numberB;
```

Java string valueOf()

- The **java string valueOf()** method converts different types of values into string.

- By the help of string valueOf() method, you can convert int, long, boolean, character, float, double, object and char array to string

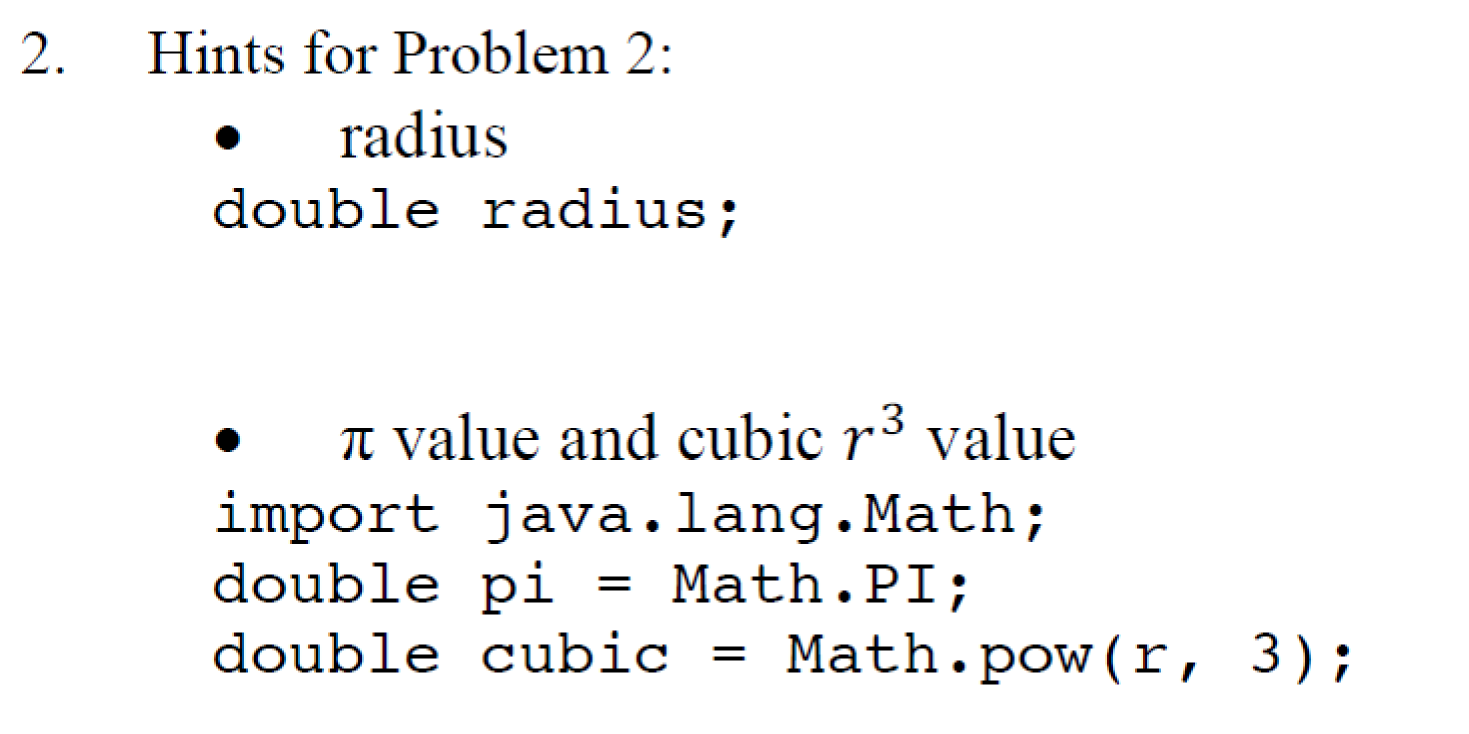
Tutorial Q2



2.Write a program that reads the radius of a sphere and prints its volume and surface area. Use the following formulas, where r represents the radius:

(a) Volume $= \frac{4}{3}\pi r^3$

(b) Surface Area $= 4\pi r^2$



2. Hints for Problem 2:

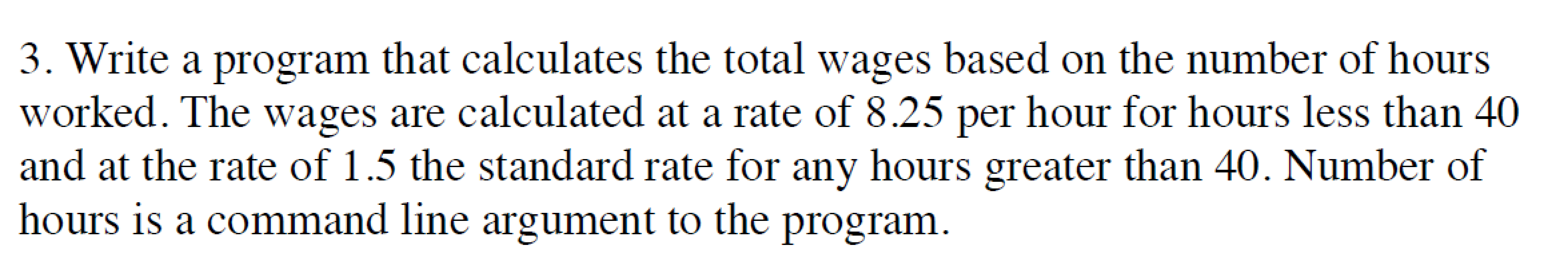
- radius

```
double radius;
```

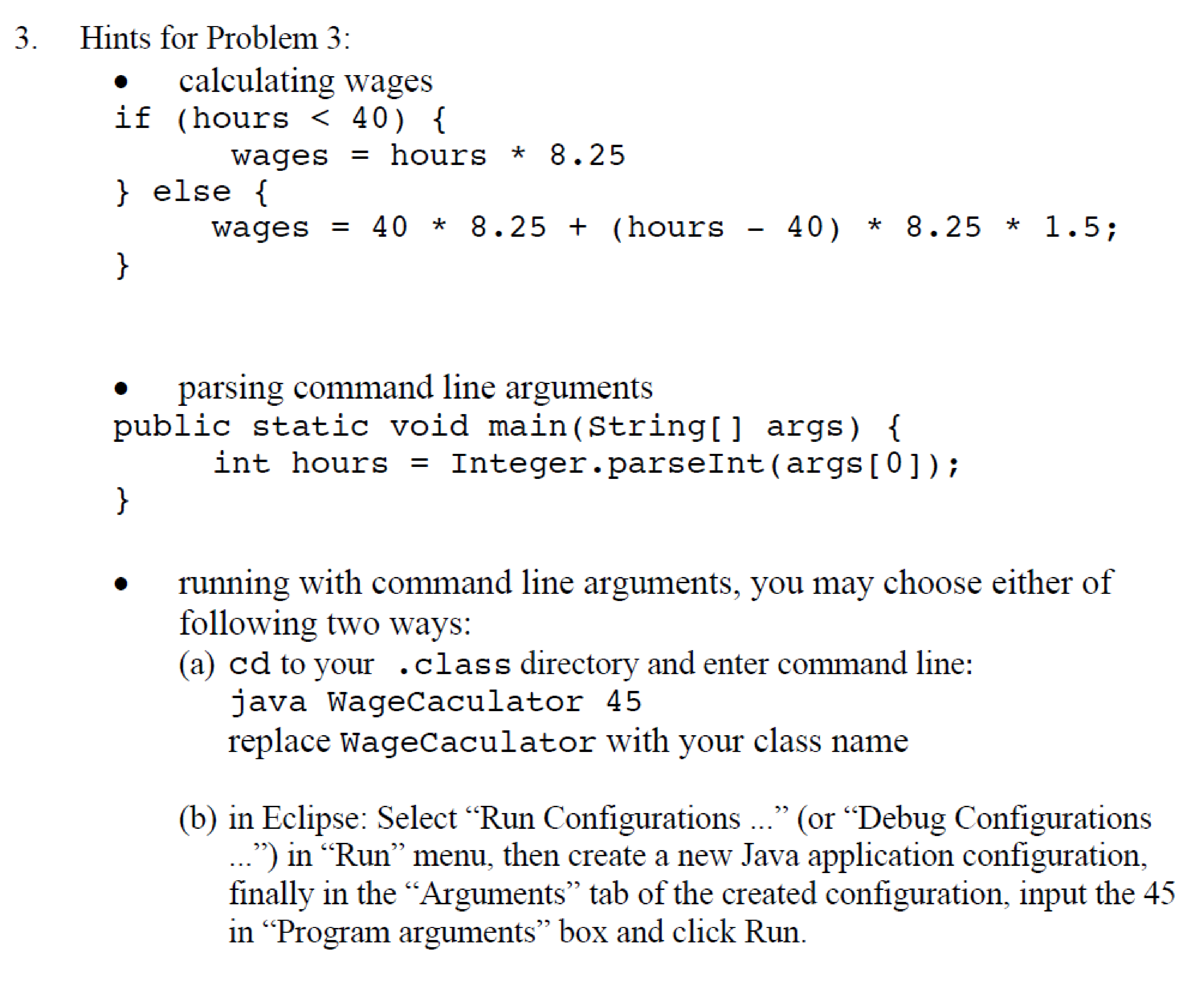
- π value and cubic r^3 value

```
import java.lang.Math;
double pi = Math.PI;
double cubic = Math.pow(r, 3);
```

Tutorial Q3



3. Write a program that calculates the total wages based on the number of hours worked. The wages are calculated at a rate of 8.25 per hour for hours less than 40 and at the rate of 1.5 the standard rate for any hours greater than 40. Number of hours is a command line argument to the program.



3. Hints for Problem 3:

- calculating wages

```
if (hours < 40) {
      wages = hours * 8.25
} else {
     wages = 40 * 8.25 + (hours – 40) * 8.25 * 1.5;
}
```

- parsing command line arguments

```
public static void main(String[] args) {
     int hours = Integer.parseInt(args[0]);
}
```

- running with command line arguments, you may choose either of following two ways:

  (a) `cd` to your `.class` directory and enter command line:
  `java WageCaculator 45`
  replace `WageCaculator` with your class name

  (b) in Eclipse: Select “Run Configurations ...” (or “Debug Configurations ...”) in “Run” menu, then create a new Java application configuration, finally in the “Arguments” tab of the created configuration, input the 45 in “Program arguments” box and click Run.

The Conditional Operator

- The conditional operator is a notational variant on certain forms of the if-else statement
    - Also called the ternary operator or arithmetic if
    - The following examples are equivalent:

```
if (n1 > n2) max = n1;
else max = n2;
```

vs.

```
max = (n1 > n2) ? n1 : n2;
```

- The expression to the right of the assignment operator is a conditional operator expression
- If the Boolean expression is **True** → the first expression (n1)
  Else If the Boolean expression is **False** → second expression (n2)