

q1 Student number: _____ Last Name: _____ First Name: _____

```
class Integer:          | class Double:
  // = s's value, as an int.    | // = s's value, as a double
  static int parseInt(String s) | static double parseDouble(String s)
```

```
class String:
  String substring(int i, int j) // = the letters from i (inclusive) to j (non-inclusive)
  String substring(int i) // = the letters from i (inclusive) to the end
  int indexOf(String s) // = the index of s in this String; -1 if s is not a substring
  int indexOf(String s, int i) // = index of s in this String after i; -1 if s not found
  int length() // = the number of characters in this String
  boolean equals(String s) // = this String has the same contents as s
```

```
class JOptionPane:
  // Display 'message' in a message dialog; c is usually null in CSC108H
  static void showMessageDialog(Component c, Object message)
  // Display 'message' in an input dialog, and return what the user types
  static String showInputDialog(Object message)
```

```
class BufferedReader:
  // Create a reader reading from 'in' (usually a FileReader in CSC108H)
  BufferedReader(Reader in)
  // Return the next available line in the BufferedReader, or null if at end
  String readLine()
```

```
class FileReader:
  FileReader(String f) // Create a reader reading from a file named f
```

```
class Math:
  static double abs(double a) // = the absolute value of a
  static int abs(int a) // = the absolute value of a
  static double pow(double a, double b) // = a^b
```

1. [4 marks] Complete the method below.

```
/**
 * Prompt the user for two Strings (with a separate input dialog
 * window for each) and display the shorter one (in a message dialog).
 * Display either if they are the same length.
 */
public static void showShorter() {
  String s1= JOptionPane.showInputDialog("enter a string:");
  String s2= JOptionPane.showInputDialog("enter another string:");
  if (s1.length() < s2.length()) {
    JOptionPane.showMessageDialog(null, s1);
  }
  else {
    JOptionPane.showMessageDialog(null, s2);
  }
}
```

2. [6 marks] Complete the following method.

```
/**
 * Read a sequence of Strings from the file named f and return whether s
 * occurs as a line in f.
 */
public static boolean isStringInFile(String f, String s) throws Exception {
    BufferedReader br= new BufferedReader(new FileReader(f));
    for (String in= br.readLine(); in != null; in= br.readLine()) {
        if (s.equals(in)) {
            return true;
        }
    }
    return false;
}
```

3. [4 marks] Complete the method below.

```
/**
 * Print the squares of the numbers in the range i..j (inclusive). That is,
 * print  $i^2$   $(i+1)^2$  ...  $j^2$ , with each number on a line by itself.
 */
public static void printSquaresInRange(int i, int j) {
    for (int k= i, k <= j; k++) {
        System.out.println(k*k);
    }
}
```

4. [4 marks] Complete the following method.

```
/**
 * Print the characters of s in reverse order, one per line.
 * Precondition: s != null.
 */
public static void print(String s) {
    for (int i= s.length() - 1; i != -1; i--) {
        System.out.println(s.charAt(i));
    }
}
```

q2 Student number: _____ Last Name: _____ First Name: _____

```
class Integer:                | class Double:
// = s's value, as an int.    | // = s's value, as a double
static int parseInt(String s) | static double parseDouble(String s)

class String:
String substring(int i, int j) // = the letters from i (inclusive) to j (non-inclusive)
String substring(int i) // = the letters from i (inclusive) to the end
int indexOf(String s) // = the index of s in this String; -1 if s is not a substring
int indexOf(String s, int i) // = index of s in this String after i; -1 if s not found
int length() // = the number of characters in this String
boolean equals(String s) // = this String has the same contents as s

class JOptionPane:
// Display 'message' in a message dialog; c is usually null in CSC108H
static void showMessageDialog(Component c, Object message)
// Display 'message' in an input dialog, and return what the user types
static String showInputDialog(Object message)

class BufferedReader:
// Create a reader reading from 'in' (usually a FileReader in CSC108H)
BufferedReader(Reader in)
// Return the next available line in the BufferedReader, or null if at end
String readLine()

class FileReader:
FileReader(String f) // Create a reader reading from a file named f

class Math:
static double abs(double a) // = the absolute value of a
static int abs(int a) // = the absolute value of a
static double pow(double a, double b) // = a^b
```

1. [4 marks] Complete the method below.

```
/**
 * Prompt the user for two integers (with a separate input dialog
 * window for each) and display the smaller one (in a message dialog).
 * Display either if they are the same.
 */
public static void showSmaller() {
    int i1= Integer.parseInt(JOptionPane.showInputDialog("type an integer:"));
    int i2= Integer.parseInt(JOptionPane.showInputDialog("type another integer:"));
    if (i1 < i2) {
        JOptionPane.showMessageDialog(null, " + i1);
    }
    else {
        JOptionPane.showMessageDialog(null, " + i2);
    }
}
```

2. [6 marks] Complete the following method.

```
/**
 * Read a sequence of doubles from the file named f and
 * return the average.
 */
public static double averageNumsInFile(String f) throws Exception {
    int numDoubles= 0;
    double sumDoubles= 0;
    BufferedReader br= new BufferedReader(new FileReader(f));
    for (String in= br.readLine(); in != null; in= br.readLine()) {
        sum += Double.parseDouble(in);
        numDoubles++;
    }
    if (numDoubles == 0) {
        return 0.0;
    }
    else {
        return sumDoubles / numDoubles;
    }
}
```

3. [4 marks] Complete the following method.

```
/**
 * Print the characters of s separated by spaces. (it's okay to have an
 * extra space at the end).
 * Precondition: s != null.
 */
public static void print(String s) {
    for (int i= 0; i != s.length(); i++) {
        System.out.print(s.charAt(i) + " ");
    }
}
```

4. [4 marks] Complete the method below.

```
/**
 * Print the cubes of the numbers in the range i..j (inclusive). That is,
 * print  $i^3$   $(i+1)^3$  ...  $j^3$ , with each number on a line by itself.
 */
public static void printCubesInRange(int i, int j) {
    for (int k= i; k <= j; k++) {
        System.out.println(k*k*k);
    }
}
```