Homework One

A201/A507/I210
Spring Semester 2005

Due in writing at the end of lecture on Jan. 27

Abstract

Write programs that solve the problems below as indicated. Programs should be submitted to OnCourse in the dropbox for Homework One. A report including a very detailed explanation for each program describing the design and implementation as well as the rationale behind your solution and/or approach is to be turned in at the time of the lecture on Thursday, January 27. You’re encouraged to work in groups and discuss the problems but you need to write the programs and prepare the write-up all by yourself. As an analogy assume you’re sportswriters\(^1\) for one of the major newspapers in the business (Sports Illustrated, New York Times, The Chicago Tribune, IDS\(^2\)) and you are allowed to watch a game of basketball together\(^3\), and discuss it with each other, but when you write your report for the newspaper you’re working for, you each write your own, from scratch, and trying to be as original and authentic as you could possibly be. Otherwise the newspaper will terminate your employment. In our case the Computer Science Department\(^4\) and the School of Informatics \(^5\) clearly specify the rules of academic honesty and academic integrity, so please read the documents and make sure you understand them and comply with them.

Also posting solutions or major hints on the bulletin board is not allowed.

1 The Temperature Converter

Write a program that converts Fahrenheit temperatures into Celsius. Assuming that \(f\) is the given temperature in degrees Fahrenheit, the temperature in degrees Celsius can be calculated using the following formula:

\[
c = \frac{5}{9} \cdot (f - 32)
\]

\(^1\) Or, more general, just “pundits.”
\(^2\) You name it.
\(^3\) Same goes for a presidential debate, for example.
\(^4\) http://www.cs.indiana.edu/academics/integrity.html
\(^5\) http://www.informatics.indiana.edu/courses/honesty.asp
2 Calculating A Car’s Autonomy

Given a car’s mileage \( m \) (expressed in miles per gallon) and the amount of fuel in the car’s tank \( f \) (expressed in gallons), write a formula to determine the autonomy of that car in miles. Then use the formula that you wrote to write a program that asks the user for the mileage of a car and the amount of fuel in the car and prints back the autonomy of the car in miles. When finished your program should behave like this:

```java
frilled.cs.indiana.edu%javac Autonomy.java
frilled.cs.indiana.edu%java Autonomy
Please enter the mileage for this car (miles/gallon): 20.7
Please enter the amount of fuel (in gallons): 2.6
The autonomy of the car is: 53.82 miles.
frilled.cs.indiana.edu%javac Autonomy.java
frilled.cs.indiana.edu%java Autonomy
Please enter the mileage for this car (miles/gallon): 50
Please enter the amount of fuel (in gallons): 0
The autonomy of the car is: 0.0 miles.
frilled.cs.indiana.edu%javac Autonomy.java
frilled.cs.indiana.edu%java Autonomy
Please enter the mileage for this car (miles/gallon): 50
Please enter the amount of fuel (in gallons): -2
The autonomy of the car is: -100.0 miles.
frilled.cs.indiana.edu%javac Autonomy.java
frilled.cs.indiana.edu%java Autonomy
Please enter the mileage for this car (miles/gallon): three
Exception in thread "main" java.lang.NumberFormatException:
For input string: "three"
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java:48)
    at java.lang.Float.parseFloat(String.java:1207)
    at java.lang.Double.parseDouble(Double.java:220)
    at ConsoleReader.parseDouble(ConsoleReader.java:26)
    at Autonomy.main(Autonomy.java:5)
```
3 Calculating Square Roots

Pages 20-21 in your text (Soliloquy) outline a method of calculating the square root of $n$ by successive approximations. Write a program that asks for a positive number $n$ and then uses the method described in the book to calculate the first ten approximations of $\sqrt{n}$ starting from the initial approximation of $x_0 = 1$. At each iteration the program should print the current approximation but also the square of that approximation to see how close to $n$ we are if we multiply the current approximation by itself.

Here's how the program might behave, when finished:

```
fripped.cs.indiana.edu\%javac SquareRoot.java
fripped.cs.indiana.edu\%java SquareRoot
Please enter a positive number: 5
We're going to calculate successive approximations of the square root of 5.0 now.
The first approximation is 1.0,
... and 1.0 squared is: 1.0
The second approximation is 3.0,
... and 3.0 squared is: 9.0
The third approximation is 2.33333333333335,
... and 2.33333333333335 squared is: 5.4444444444444455
The fourth approximation is 2.238095238095238
... and 2.238095238095238 squared is: 5.09070294794581
The fifth approximation is 2.360688956433634
... and 2.360688956433634 squared is: 5.500004106063731
The sixth approximation is 2.3606797749997978
... and 2.36067977499978 squared is: 5.00000000000843
The seventh approximation is 2.2360797749997979
... and 2.23607977499979 squared is: 5.00000000000001
The eighth approximation is 2.2360797749997979
... and 2.23607977499979 squared is: 5.00000000000001
The ninth approximation is 2.2360797749997979
... and 2.23607977499979 squared is: 5.00000000000001
The tenth approximation is 2.2360797749997979
... and 2.23607977499979 squared is: 5.00000000000001
```

```
fripped.cs.indiana.edu\%javac SquareRoot.java
fripped.cs.indiana.edu\%java SquareRoot
Please enter a positive number: 175
We're going to calculate successive approximations of the square root of 175.0 now.
The first approximation is 1.0,
... and 1.0 squared is: 1.0
The second approximation is 88.0,
... and 88.0 squared is: 7744.0
The third approximation is 44.99431818181818,
... and 44.994318181818 squared is: 2024.4886686466941
The fourth approximation is 24.441849077018446
... and 24.441849077018446 squared is: 597.4039863037475
The fifth approximation is 15.800850088506676
... and 15.800850088506676 squared is: 249.66686351946143
```
The sixth approximation is 13.43810178378815,
... and 13.43810178378815 squared is: 180.58257955145027
The seventh approximation is 13.230387195773005,
... and 13.230387195773005 squared is: 175.0431453600743
The eighth approximation is 13.228756655810876,
... and 13.228756655810876 squared is: 175.00000265866058
The ninth approximation is 13.228756555322953,
... and 13.228756555322953 squared is: 175.0
The tenth approximation is 13.228756555322953,
... and 13.228756555322953 squared is: 175.0
frilled.cs.indiana.edu%

4 An Investment Problem

Pages 41-42 in your text describe an investment problem and its solution (in pseudocode). Write a program to solve a similar problem as described below. Assume that your bank account starts with m amount of dollars in it and you never take money of your account. At the end of the year the amount in your account grows by 5%. Write a program that asks the user for the initial amount m, then shows what happens to the account for ten straight years. When finished your program might work like this:

frilled.cs.indiana.edu%javac Investment.java
frilled.cs.indiana.edu%java Investment
Please enter the initial balance (in dollars): 10
After a year the balance becomes: 10.5
After two years the balance becomes: 11.025
After three years the balance becomes: 11.57625
After four years the balance becomes: 12.150625
After five years the balance becomes: 12.762815625
After six years the balance becomes: 13.40095640625
After seven years the balance becomes: 14.0710042265625
After eight years the balance becomes: 14.774554437890625
After nine years the balance becomes: 15.513282159785156
After ten straight years the balance becomes: 16.28894626777413 dollars.
frilled.cs.indiana.edu%javac Investment.java
frilled.cs.indiana.edu%java Investment
Please enter the initial balance (in dollars): 25000
After a year the balance becomes: 26250.0
After two years the balance becomes: 27562.5
After three years the balance becomes: 28940.625
After four years the balance becomes: 30387.65625
After five years the balance becomes: 31907.0390625
After six years the balance becomes: 33502.39105625
After seven years the balance becomes: 35177.51056640625
After eight years the balance becomes: 36936.386094726564
After nine years the balance becomes: 38783.205399462895
After ten straight years the balance becomes: 40722.36566943604 dollars.
5 Reversing Strings

Write a program that reverses strings of characters. The program should ask the user for a string of up to 10 characters and then print the string reversed. When finished your program might work as shown below. Strings of no characters should be processed correctly by your program. What the program should do when the input is more than 10 characters long is unspecified, that is, up to you, but please note that the example below offers a suggestion, which you could use:

```java
frilled.cs.indiana.edu%javac Reverse.java
frilled.cs.indiana.edu%java Reverse
Please enter a string of at most 10 characters: Adrian
You have entered the string: Adrian
Adrian reversed is naird
frilled.cs.indiana.edu%javac Reverse
Please enter a string of at most 10 characters: One
You have entered the string: One
One reversed is en0
frilled.cs.indiana.edu%javac Reverse
Please enter a string of at most 10 characters: Schwartzbenck
You have entered the string: Schwartzbenck
Schwartzbenck reversed is kcebnmeztra
frilled.cs.indiana.edu%javac Reverse
Please enter a string of at most 10 characters:
You have entered the string: reversed is
frilled.cs.indiana.edu%javac Reverse
Please enter a string of at most 10 characters: a
You have entered the string: a
a reversed is a
frilled.cs.indiana.edu%javac Reverse
Please enter a string of at most 10 characters: pooh
You have entered the string: pooh
pooh reversed is hoop
frilled.cs.indiana.edu%`
```

6 Swapping First with Last in a String

Write a program that asks for a string of characters of any length and writes back the string obtained by swapping the first with the last characters in the given string. Here’s how your program might behave when finished, including two situations when the behaviour of the program is unspecified (so the program could throw a legitimate error in those cases):

```java
frilled.cs.indiana.edu%javac Swap.java
frilled.cs.indiana.edu%java Swap
Please enter a string of characters (of any length): alpha
Here’s the string after the swap: alpha
frilled.cs.indiana.edu%java Swap
```
Please enter a string of characters (of any length): hoop
Here's the string after the swap: pooh
frilled.cs.indiana.edu%java Swap
Please enter a string of characters (of any length): abcde
Here's the string after the swap: ebcda
frilled.cs.indiana.edu%java Swap
Please enter a string of characters (of any length): a
Exception in thread "main" java.lang.StringIndexOutOfBoundsException:
String index out of range: -1
  at java.lang.String.substring(String.java:1444)
  at Swap.main(Swap.java:6)
frilled.cs.indiana.edu%java Swap
Please enter a string of characters (of any length):
Exception in thread "main" java.lang.StringIndexOutOfBoundsException:
String index out of range: -1
  at java.lang.String.charAt(String.java:444)
  at Swap.main(Swap.java:6)
frilled.cs.indiana.edu%java Swap
Please enter a string of characters (of any length): ab
Here's the string after the swap: ba
frilled.cs.indiana.edu%

7 Calculating the Area of a Triangle

The area of a triangle can be calculated from its side lengths by Heron's formula:

\[ \text{Area} = \sqrt{s(s-a)(s-b)(s-c)} \]

where \(a\), \(b\), and \(c\) are the side lengths and \(s\) is the semiperimeter:

\[ s = \frac{a + b + c}{2} \]

Write a program to read the three sides of a triangle and calculate its area
by Heron's formula. When finished your program might behave like this:

frilled.cs.indiana.edu%java Triangle
Please enter the length of the first side: 3
Please enter the length of the second side: 4
Please enter the length of the third side: 5
The semiperimeter is: 6.0
The area of the triangle is: 6.0
frilled.cs.indiana.edu%java Triangle
Please enter the length of the first side: 1
Please enter the length of the second side: 1
Please enter the length of the third side: 1.4142
The semiperimeter is: 1.7071
The area of the triangle is: 0.49999999990803207
frilled.cs.indiana.edu%java Triangle
Please enter the length of the first side:
Exception in thread "main" java.lang.NumberFormatException: empty String
at java.lang.FloatinDecimal.readJavaFormatString(FloatinDecimal.java:983)
at java.lang.Double.parseDouble(Double.java:220)
at ConsoleReader.parseDouble(ConsoleReader.java:26)
at Triangle.main(Triangle.java:5)
frilled.cs.indiana.edu%java Triangle
Please enter the length of the first side: -2
Please enter the length of the second side: 3
Please enter the length of the third side: 4
The semiperimeter is: 2.5
The area of the triangle is: 2.90473750965555625
frilled.cs.indiana.edu%

Cases are included where the behaviour of the program is unspecified. In other words instead of a user-friendly program we’re talking of a program-friendly user.

8 Extracting Substrings

Write a program that asks the user for a string (of any length) and two numbers: two positive integers $i$ and $j$, such that $i \leq j$. The program is to print back all the characters in the string that appear after (and excluding) the first $i$ characters in the string, in the same order, and up to (and including) the $j$-th character in the string. Here’s how the program should behave when finished:

frilled.cs.indiana.edu%java Extract
Please enter a string: something
Please enter the first integer: 3
Please enter the second integer: 4
The extracted string is: e
frilled.cs.indiana.edu%java Extract
Please enter a string: something
Please enter the first integer: 0
Please enter the second integer: 5
The extracted string is: somet
frilled.cs.indiana.edu%java Extract
Please enter a string: automatic
Please enter the first integer: 2
Please enter the second integer: 8
The extracted string is: tomato
frilled.cs.indiana.edu%java Extract
Please enter a string: here’s a longer string
Please enter the first integer: 4
Please enter the second integer: 11
The extracted string is: ’s a lo
frilled.cs.indiana.edu%java Extract
Please enter a string: nothing
Please enter the first integer: 6
Please enter the second integer: 2
Exception in thread "main" java.lang.StringIndexOutOfBoundsException:
String index out of range: -4  
    at java.lang.String.substring(String.java:1444)  
    at Extract.main(Extract.java:10)
frilled.cs.indiana.edu%java Extract
Please enter a string:
Please enter the first integer: 3
Please enter the second integer: 5
Exception in thread "main" java.lang.StringIndexOutOfBoundsException:
String index out of range: 5  
    at java.lang.String.substring(String.java:1441)  
    at Extract.main(Extract.java:10)
frilled.cs.indiana.edu%

9 Euclidean Distance in the Plane

The Euclidean distance between two points \((x_1, y_1)\) and \((x_2, y_2)\) in the plane is:

\[
\text{Distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}
\]

Write a program that reads the coordinates of two points and calculates then distance between them. When finished your program might work like the one illustrated below:

frilled.cs.indiana.edu%java Distance
Please enter \(x_1\): 0
Please enter \(y_1\): 3
Please enter \(x_2\): 4
Please enter \(y_2\): 0
The distance between (0.0, 3.0) and (4.0, 0.0) is: 5.0
frilled.cs.indiana.edu%java Distance
Please enter \(x_1\): -1
Please enter \(y_1\): 4
Please enter \(x_2\): -3.2
Please enter \(y_2\): 18.9
The distance between (-1.0, 4.0) and (-3.2, 18.9) is: 15.06154042586614
frilled.cs.indiana.edu%

10 Calculating a Test

Knowing that \(\text{Math.abs}(x)\) can be used to calculate the absolute value of a variable \(x\) write a program that asks the user for two numbers then
prints the largest of the two back\(^6\). So here’s how my program works:

```java
frilled.cs.indiana.edu%java Max
Please enter the first number: 1
Please enter the second number: 3
3 is the largest of 1 and 3
frilled.cs.indiana.edu%java Max
Please enter the first number: 3
Please enter the second number: 1
3 is the largest of 3 and 1
frilled.cs.indiana.edu%java Max
Please enter the first number: -1
Please enter the second number: -3
-1 is the largest of -1 and -3
frilled.cs.indiana.edu%java Max
Please enter the first number: -3
Please enter the second number: -1
-1 is the largest of -3 and -1
frilled.cs.indiana.edu%
```

Behaviour not shown up is up to you (choice is yours).
Also, the use of `Math.max(x, y)` is explicitly prohibited.

\(^6\)Hopefully this note won’t mean much for anybody, but just to be sure: the use of if is prohibited. We have not even studied it yet.