Hands #0 - Getting Started with Input/Output and printf

Hands #0:

How to accomplish common tasks:

Reading input from the keyboard
Creating data files
Reading input from data files
Using `split` to split strings into array elements for parsing
Parsing integers from strings
Reading an unknown amount of data from a data file
Using `printf` to format output by rounding and spacing in columns

Creating a program for contest:

Set up a shell to read the number of test cases and write a loop to test the cases
Write the code for each test case

Fundamentals of Java 5.0 - Common Tasks

`Scanner` class - How to read input from the keyboard:

```
import java.io.*;
import java.util.*;
import static java.lang.System.*;          // don't need System before in or out.println

public class Ex1_Scanner_Keyboard {
   public static void main(String args[]) throws IOException {
      Scanner kb = new Scanner(in);
      System.out.print("Enter an integer: ");
      int a = kb.nextInt();   // reads an integer from keyboard
      System.out.println(a);

      System.out.print("Enter a double: ");
      double d = kb.nextDouble();   // reads a double from keyboard
      System.out.println(d);

      System.out.print("Enter a word: ");
      String s = kb.next();   // reads one string with no spaces from keyboard
      System.out.println(s);
   }
}
```

`Sample Run` for `Ex1_Scanner_Keyboard`

```
Enter an integer: 12
12
Enter a double: 14
14.0
Enter a word: Java
Java
```

How to create an input file:

1. Open a new file.
2. Enter data for your input file.
3. Save as *filename*`.dat` or *filename*`.txt`

Note: BE SURE to save your data file in the same folder with your source code

`Scanner class` - How to read input from a data file:

```
import java.io.*;
import java.util.*;
import static java.lang.System.*;

public class Ex2_Scanner_InputFile {
   public static void main(String args[]) throws IOException {
      Scanner in = new Scanner(new File("Ex2.dat"));
      // read a single integer from a dat file
      String s = in.nextLine().trim();     // reads the first line of the dat file, the 3, as a String
      int num = Integer.parseInt(s);       // convert s to an int

      // read several integers from a single line
      // reads a line of  dat file, the 5 6 as a String
      // splits the String around the spaces into the array {"48", "16"}
      String [] t = in.nextLine().trim().split(" ");

      // convert the elements of the array to ints
      int x = Integer.parseInt(t[0]);
      int y = Integer.parseInt(t[1]);

      // now you can use the integers as integers in formulas
      int sum = x + y;
      out.println("Sum = " + sum);
   }
}
```

Ex2.dat

```
3
48 16
```

Output

```
Sum = 64
```

`Scanner` class - How to read an unknown amount of input from a data file using loops

```
import java.io.*;
import java.util.*;
import static java.lang.System.*;

public class Ex3_Scanner_Loops {
   public static void main(String args[]) throws IOException {
      Scanner in = new Scanner(new File("Ex3.dat"));
      while(in.hasNext()) {
         String s = in.nextLine().trim();     //reads a string from input file
         System.out.println(s);
      }
   }
}
```

Ex3.dat

```
34
48 16
Dog
Cat
3 4 5 a b c
```

output

```
34
48 16
Dog
Cat
3 4 5 a b c
```

How to output using `printf`

statements in the form: `out.printf("some string with %code for each arg", arg1, arg2,...)`

`printf` codes:

`%d` = int base 10	`%s` = String
`%f` = double	`%o` = octal
`%e` = scientific	`%x` = hexadecimal
`%c` = char	

How to format using `printf` (works for Strings, too)

`%.2f` round `double` to two decimals
`%5.1f` round `double` to one decimal and **right** justify it in a string of 5 characters
`%-5.1f` round `double` to one decimal and **left** justify it in a string of 5 characters
`%05.1f` round `double` to one decimal and **right** justify it in a string of 5 characters and put 0 in empty spaces

```
import java.io.*;
import java.util.*;
import static java.lang.System.*;
public class Ex4_printf {
   public static void main(String args[]) throws IOException  {

      String name = "Java is fun";
      int id = 1111;
      double gpa = 4.15;

      // %s is where to print the variable name
      out.printf("%s is a wonderful person\n", name); // args can be variables or literals
      out.printf("ID number %d, name %s has GPA: %f \n\n", id, name, gpa);
      out.printf("ID number %d, name %s has GPA: %f \n\n", 222, "Liberace", 3.2);

      double sam = 234.56789;
      out.printf("1 decimal:   %.1f \n", sam);  // round to one decimal
      out.printf("Left justify in 6 columns%-6.0frounded to integer\n\n", sam);
      // 8.3 means right justified in 8 spaces with a decimal and 3 decimal digits
      out.printf("3 decimals:   %8.3f \n\n", sam);

      // output in scientific notation
      out.printf("\nScientific notation:   %e \n\n", sam); //floating point output

      // output in chars
      out.printf("\nThese are A's %c %c \n\n", 'a', 65); // can be a variable or ASCII

      int num = 640;

      // print in different bases
      out.printf ("  base 10: %d \n", num);     // decimal - base 10
      out.printf("  base  8: %o \n", num);      // octal - base 8
      out.printf("  base 16: %x \n\n", num);    // hexadecimal - base 16

      // format String output
      out.printf("%10d%s \n", 111, "Right justify in 10 columns");
      out.printf("%010d%s \n", 111, "Right justify and fill with 0's");
      out.printf("%-10d%s \n", 111, "Left justify in 10 columns");
   }
}
```

Output is on the next page.

Output for `printf` code on previous page:

```
Java is fun is a wonderful person
ID number 1111, name Java is fun has GPA: 4.150000

ID number 222, name Liberace has GPA: 3.200000

1 decimal:   234.6

Left justify in 6 columns235   rounded to integer
3 decimals:    234.568


Scientific notation:   2.345679e+02


These are A's a A

  base 10: 640
  base  8: 1200
  base 16: 280

       111Right justify in 10 columns
0000000111Right justify and fill with 0's
111       Left justify in 10 columns
```

Template for Solving Hands-on Problems

Solutions for most problems in the hands-on contest have the following format:

1. Read in the number of test cases

2. For each test case
   - Read in the data to be tested
   - Output the results

Sample Problem - pr01

Most contest problems are written in a style similar to this problem:

Problem:	Write a program that will find the perimeter of a quadrilateral, given the lengths of its sides.
Input:	The first line of the data set is an integer that represents the number of lines that follow. Each of the remaining lines contains the lengths of the sides of one rectangle.
data file:	`pr01`
Output:	Output the label "`PERIMETER: `" followed by the perimeter of the quadrilateral.
Assumptions:	All data sets will form a quadrilateral.

Sample Input:	`3`
	`3 4 5 6`
	`4 5 4 5`
	`6 6 6 6`

Output Input:	`PERIMETER: 18`
	`PERIMETER: 18`
	`PERIMETER: 24`

Steps for creating the solution to this problem is on the next page but first you need a `dat` file.

How to Create a dat file:

1. type the data into a file in JCreator (or whatever IDE you use), a word processor, or notepad.
2. save it as `pr##.dat` or `pr##.txt` or whatever you are told for the contest
3. place it in the same folder as your program.

Steps for creating a solution to Sample pr01

Step 1: Create a shell with import statements and loop to test all the cases (have students use same `Scanner` and `counter` variables on all programs so they don't have to think through new names all the time)

```
import java.util.*;
import java.io.*;
import static java.lang.System.*;

public class pr01
{
   public static void main (String[] args) throws IOException
   {
      Scanner kb = new Scanner(new File("pr01.dat"));
      int times = Integer.parseInt(kb.nextLine().trim());      // *1
      for(int z = 0; z < times; z++) {

         // this is where you write the code for to solve the problem
      }
   }
}
```

***1** parses the `String` received from `kb.nextLine()` to an int.

Efficiency suggestions for writing the next program:

1. Make the shell as soon as the contest begins with as many `import` statements as needed.
2. Save it as "`shell.java`" in a folder for that contest
3. Resave it as the first problem you are working and then fill in the `for` loop.
4. Be sure to change the class name and the dat file name
5. Repeat for other problems.

Step 2: Now it is time to add the code inside the `for` loop to find the perimeter of each rectangle.

```
import java.util.*;                                                    pr01
import java.io.*;
import static java.lang.System.*;

public class pr01
{
   public static void main (String[] args) throws IOException
   {
      Scanner kb = new Scanner(new File("pr01.dat"));
      int times = Integer.parseInt(kb.nextLine().trim());
      for(int z = 0; z < times; z++) {
         String[] s = kb.nextLine().trim().split(" ");    // *2
         int sum = 0;
         for(int i=0; i<4; i++)
               sum += Integer.parseInt(s[i]);             // *3
         out.println("PERIMETER = " + sum);
      }
   }
}
```

```
3
3 4 5 6
4 5 4 5
6 6 6 6
```

output

```
PERIMETER = 18
PERIMETER = 18
PERIMETER = 24
```

*2 this code reads a line. In this example: "3 4 5 6" is the read the first time through the loop.
`trim()` trims extra white space, including end of line markers
Then the string is split around the spaces into an array of strings:
`s[0]="3". s[1]="4". s[2]="5". s[3]="6"`

*3 this code parses the `String s[i]` to an `int`.