Homework 1

- Ungraded, but I have gone over them anyhow
- In order to get a 5, your program had to
  5.1 Be syntactically correct - it compiled
  5.2 Have proper comments including names of both partners, date, assignment number and program description in opening comment, and at least one useful inline comment.
  5.3 Have proper and consistent indenting.
  5.4 Have descriptive variable names (e.g. data1 not x for the first number read in).
  5.5 Produce correct output.
  5.6 Have descriptive, well formatted output.

Homework 1

Common problems
- Header comment incomplete
- No inline comments
- Improper indenting
- Insufficient whitespace - use whitespace to separate statements into related groups
- Unnecessary recalculation

Next Homework

- Homework 2 due Wednesday, Jan 22 - graded
- Homework 3 due Monday, Jan 27 - ungraded
- Homework 4 due Friday, Jan 31 - graded
- Make sure you have a partner!
- Don’t forget to use the time logs.

<table>
<thead>
<tr>
<th>Homework 1</th>
<th>Last Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static void main( String[] args ) { double num1, num2, num3, sum, average; System.out.println(&quot;Enter a number.&quot;); num1 = Console.in.readDouble(); //get first number System.out.println(&quot;You entered &quot;+num1); System.out.println(&quot;The sum of the two numbers is &quot;); sum=num1+num2; // add the first two numbers together System.out.println(sum); System.out.println(average); System.out.println(average); System.out.println(&quot;The average of the two numbers is &quot;); average=(num1+num2)/2; System.out.println(average); System.out.println(average); System.out.println(&quot;The average of the three numbers is &quot;); average=(num1+num2+num3)/3; System.out.println(average); System.out.println(average); System.out.println(average);</td>
<td></td>
</tr>
<tr>
<td>boolean b = ((age &gt;= 18) &amp;&amp; (age &lt; 65)); System.out.println(&quot;Full Fare Adult: &quot;+b); b = ((age &lt; 18)</td>
<td></td>
</tr>
<tr>
<td>We wanted to rewrite this code: boolean b = ((age &gt;= 18) &amp;&amp; (age &lt; 65)); System.out.println(&quot;Full Fare Adult: &quot;+b); b = ((age &lt; 18)</td>
<td></td>
</tr>
</tbody>
</table>
| To use if to print out an appropriate message:
'Full Fare Adult' if age is between 18 and 65
'Reduced Fare' if age < 18 or age > 65
| And we produced this code:
// Check for age between 18 and 65 if ( age >= 18 ) && ( age <= 65 ) { System.out.println("Full Fare Adult"); } // Check for younger than 18 or older than 65 if ( age < 18 ) || ( age > 65 ) System.out.println("Reduced Fare"); |
Question
- The original code used a boolean variable b. Why didn’t we need one in the code that we developed?
  - The original example:
    ```java
    boolean b = ((age >= 18) && (age < 65));
    System.out.println("Full Fare Adult: " + b);
    b = ((age < 18) || (age >= 65));
    System.out.println("Eligible for reduced fare: " + b);
    
    - Our new code:
      // Check for age between 18 and 65
      if ( (age >= 18) && (age <= 65) ) {
        System.out.println("Full Fare Adult");
      } // Check for younger than 18 or older than 65
      if ( (age <18) || (age > 65))
        System.out.println("Reduced Fare");
    ```

If in action - Example: Bubblesort
- Given three numbers, place them in increasing order
- Algorithm
  1. Put the three numbers in a, b, and c
  2. If b is less than a, swap a and b
  3. If c is less than b,
     1. swap b and c
     2. If b is less than a, swap a and b

Bubblesort (1)

1. Put the three numbers in a, b, and c
2. If b is less than a, swap a and b
3. If c is less than b,
   1. swap b and c
   2. If b is less than a, swap a and b

Bubblesort (1)

1. Put the three numbers in a, b, and c
2. If b is less than a, swap a and b
3. If c is less than b,
   1. swap b and c
   2. If b is less than a, swap a and b
Bubble sort steps:

1. Put the three numbers in \(a\), \(b\), and \(c\).
2. If \(b\) is less than \(a\), swap \(a\) and \(b\).
3. If \(c\) is less than \(b\),
   1. swap \(b\) and \(c\).
   2. If \(b\) is less than \(a\), swap \(a\) and \(b\).

Ex: 5, 17, 23

1. Put the three numbers in \(a\), \(b\), and \(c\).
2. If \(b\) is less than \(a\), swap \(a\) and \(b\).
3. If \(c\) is less than \(b\),
   1. swap \(b\) and \(c\).
   2. If \(b\) is less than \(a\), swap \(a\) and \(b\).

Ex: 23, 17, 5

1. Put the three numbers in \(a\), \(b\), and \(c\).
2. If \(b\) is less than \(a\), swap \(a\) and \(b\).
3. If \(c\) is less than \(b\),
   1. swap \(b\) and \(c\).
   2. If \(b\) is less than \(a\), swap \(a\) and \(b\).

Ex: 17, 23, 5

1. Put the three numbers in \(a\), \(b\), and \(c\).
2. If \(b\) is less than \(a\), swap \(a\) and \(b\).
3. If \(c\) is less than \(b\),
   1. swap \(b\) and \(c\).
   2. If \(b\) is less than \(a\), swap \(a\) and \(b\).

Ex: 5, 23, 17
Bubblesort (2)

1. Put the three numbers in a, b, and c
2. If b is less than a, swap a and b
3. If c is less than b,
   1. swap b and c
   2. If b is less than a, swap a and b

---

Swap

- How do you swap two numbers a and b?
  - Put value of a in b, and then put value of b in a doesn’t work, because value of b gets destroyed.
    
    b ← a
    a ← b    // Oops! b’s original value is lost!
  
  - Need to store value of b in a temporary location
    
    tmp ← b
    b ← a
    a ← tmp

---

// SortInput.java - sort three numbers
import tio.*; // use the package tio

class SortInput{
    public static void main (String[] args) {
        int a, b, c, temp;
        // Get three numbers from the user
        System.out.println("type three integers:");
        a = Console.in.readInt();
        b = Console.in.readInt();
        c = Console.in.readInt();
        // If b is less than a, swap a and b
        if (b < a) {
            temp = a;
            a = b;
            b = temp;
        }
    }
}
// If c is less than b, swap b and c
if (c < b) {
    temp = b;
    b = c;
    c = temp;
    // if (the new) b is less than a, swap a and b
    if (a > b) {
        temp = a;
        a = b;
        b = temp;
    }
} System.out.print("The sorted order is : "); System.out.println(a+", " + b +", " + c);

If-else Statement

- Used to choose between two alternatives
  if ( boolean expression )
  statement
  else
  statement
- If the boolean expression is true, the then statement is executed – otherwise, the else statement is executed.

Flowchart for an If-else statement

If-else examples

if ( count != 0 )
    average = sum / count;
else
    average = 0;
    System.out.println("The average is: " + average );

if ( temperature <= 32 )
    System.out.println("It’s freezing!");
else
    System.out.println("It’s not freezing.");

Exercise

- Rewrite this code to use if-else

  // Check for age between 18 and 65
  if ( (age >= 18) && (age <= 65) ) {
    System.out.println("Full Fare Adult");
  }

  // Check for younger than 18 or older than 65
  if ( (age <18) || (age > 65))
    System.out.println("Reduced Fare");

Common Errors

- What gets printed by this?

  int x = 3, y = 4;
  if ( x < y );
    System.out.println("The smaller is: " + x);
  if ( y < x );
    System.out.println("The smaller is: " + y);


**Common Errors**

- What gets printed by this?
  ```java
  int x = 3, y = 4;
  if ( x < y )
    System.out.println("The smaller is: " + x);
  if ( y < x )
    System.out.println("The smaller is: " + y);
  
  The output is
  The smaller is 3
  The smaller is 4
  
  Watch out for empty statements.
  ```

- What gets printed by this?
  ```java
  int temperature = 45;
  if ( temperature < 32 )
    System.out.print("It is now ");
    System.out.print(32 - temperature );
    System.out.println(" below freezing.");
  System.out.println("Temperature: "+ temperature);
  
  Output:
  -13 below freezing.
  Temperature: 45
  ```

**Common Errors**

- What gets printed by this?
  ```java
  int temperature = 45;
  if ( temperature < 32 ) {
    System.out.print("It is now ");
    System.out.print(32 - temperature );
    System.out.println(" below freezing.");
  }
  System.out.println("Temperature: "+ temperature);
  ```

**Nested If**

- Any statement can be used in the then or else part of an if statement
  - Including another if statement
    ```java
    if ( age >= 18 )
      if ( age < 65 )
        System.out.println("Full Fare Adult");
    ```

**If-else-if-else-if...**

- You can string if-else statements together
- Each if-else is the statement for the previous else
  ```java
  if ( age < 18 )
    System.out.println("Child Fare.");
  else if ( age < 65 )
    System.out.println("Full Fare Adult.");
  else
    System.out.println("Senior Fare.");
  ```
What does this code print?

```java
int a = 2;
int b = 2;
if (a == 1)
    if (b == 2)
        System.out.println("a was 1, b was 2.");
else
    System.out.println("a was not 1.");
```

Dangling Else

- It doesn't print anything. The else matches the second if.
  ```java
  int a = 2;
  int b = 2;
  if (a == 1)
      if (b == 2)
          System.out.println("a was 1, b was 2.");
      else
          System.out.println("a was not 1.");
  ```
- Else always matches the nearest if in the same block

Dangling Else

- Use block statements to change the if-else matching
  ```java
  int a = 2;
  int b = 2;
  if (a == 1) {
      if (b == 2)
          System.out.println("a was 1, b was 2.");
      else
          System.out.println("a was not 1.");
  ```