Algorithms, variables, inputs, outputs, data types

Writing Programs
Sept 13
Algorithms

- Algorithms are a precise statement of how to solve a problem
  - NOT a program
  - NOT written using a PL

- Write in a way that is easy for you
- Use a pencil
  - draw circles and arrows
- Be very precise
Compute area of triangle

- Algorithm?

- Now translate it into Java code
  - How get the base and height into the program?
  - How to do the calculation in Java?
  - How to render the output?
Variables

- A thing in a program that holds a value

- Declaration -- the name and type of the variable
  - `String aString;
  - Variables must be "declared" before they can be used
    - declaration occurs once

- Initialization / Assignment
  - `aString = "aaa";
    - assignment my occur many times

- Read
  - Use the value that was initialized / assigned
  - variables must be initialized before they can be read
Compute Area of Triangle

1. Get Base and Height into program

- Use "Command Line Arguments"
  - args[0] is not memorable

- Programs need to:
  - be compilable into something runnable on computer
  - be readable by people

- So rather than using args[0],
  - base = args[0];
  - height = args[1];
  - area = base * height / 2;
Problem: Data Types

- Programs need to know the "type" of each data item
  - Consider computer memory
    - Just bits: 0 and 1
    - The TYPE tells computer how to interpret the 0 and 1.
- two simplest types in Java: char and byte
  - each use 8 bits
    - 01000001
      - if a char, then 'A'
      - ASCII table
    - if a byte then 65
Data Types

- Integer aka int
  - `int base = 55;` //32 bits 00000000 00000000 00000000 00110111

- String -- a sequence of chars
  - `String base = "55";` //16 bits 00110101 00110101
Data Types "know"

- How to interpret bits
- What operations they are allowed to do
  - number types -- standard math operations between two number types
    - + - * / %
  - String
    - concatenation between two strings
      - +
      - Curiously, "string + int" works!!!
public class FunWithInts {
    public static void main(String[] args) {
        int x;
        int y = 4;
        x = y;
        y = 3;
        System.out.println(x);
        x = y + 11;
        System.out.println("x is "+x);
        x = x + 1;
        //int x = 7; this line will not compile
        //int m = 3; this line will not compile
        //int x = 5.5; this line will not compile
        int a = 5, b = 11;
        int c = a - b;
        c = a * 2;
        c = 14;
        int r = c % 3;
        int m = 18;
        int n = m / 3;
        int k = 11 / 2;
        k = 3 / 4;
    }
}

Fill in value of each variable for each line
If not initialized ":-", not declared "x"

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>m</th>
<th>c</th>
<th>r</th>
<th>n</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>y</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rational numbers

7.1 rather than 7

- When programming "floating point numbers".
- Java floating point data type "double"
  - Usage
    - double rationalNumber = 7.1;

```java
public class GTrand {
    public static void main(String[] args) {
        double rando = Math.random();
        System.out.println(rando);
    }
}
```
Type Conversion

• String to int
  • String baseString = "5";
  • int baseInt = Integer.parseInt(baseString);

• int to String
  • int anInt = 42;
  • String aString = Integer.valueOf(anInt).toString();
  • or much shorter: String aString = ""+ anInt;

• Sometimes it is much easier, Java does it for you.
Compute area of Triangle

1. Get Base and Height into program

• From "public static void main(String[] args) {
  • know that the variable args holds strings
  • to compute area need numbers
  int base = Integer.parseInt(args[0]);
  int height = Integer.parseInt(args[1]);
  • Declaration and initialization in one step!!!
Compute area of Triangle

1. Compute Area

• from algorithm "area = (base * height) / 2"

• In Java Code?
Compute area of Triangle

3. Output

• Print to screen using System.out.println();