Announcements

• Assignment 7 questions?
  – Beware having two events that kick in at the same time!
  – Beware of infinite loops!

• What we will do today
  – Compare Alice and Java
  – Learn a little Java
  – Experiment with Java
Chap. 11 – What’s Next? Java

• Java – object-oriented programming language
  – Classes, objects, inheritance
  – Control structures (if, while)
  – Functions, methods
  – Data types (integers, doubles, strings, arrays, lists)

• Sound familiar?
Turn Alice code into Java Code

- Select Edit Preferences

- Must restart Alice
Some Data Types in Java

• integer
  – Declare and initialize
    ```java
    int value = 0; // variable is value
    ```
  – Update/modify
    ```java
    value = value + 2;
    ```

• Real numbers
  ```java
  double number = 4.5;
  number = number * 2.0; // multiply by 2
  ```

• Careful with integer operations
  ```java
  value = 6/4; // what is value?
  ```
String data type in Java

• String is a class
• Declare String variable and initialize
  String phrase = "";
  phrase = "CompSci 4";
• Convert String to array of characters
  phrase.toCharArray()
char type in Java

- char is for one character
- Note char uses single quotes, string uses double quotes

```java
char ch = 'a';
if (ch == 'a')
{
    return "found match";
}
else
{
    return "no match";
}
```
Some String member functions

- String is a class, so has member functions
  ```java
  String phrase = "CompSci 4";
  ```
- `length()` - returns number of characters in String
  ```java
  int size = phrase.length();
  ```
- `toCharArray()` – converts string to array of characters and returns the array
- `charAt(int position)` – returns the character in an array at position
  ```java
  char ch = phrase.charAt(2);
  ```
Compare two strings with equals

- Strings are more complicated than chars, cannot use == to compare strings.

```java
String one = "apple";
String two = "apfel";
String three = "apple";
if (one.equals(two) && two.equals(three))
    return "all three strings are equal";
```
Looping over a String

• Collections loop – converts the String letters to a character array and iterates over the array with ch being one character from the array each time.
• Like Alice, getting one item-from-list at a time

```java
for (char ch: letters.toCharArray())
{
    // do something here with ch
}
```

Must have Java 1.5 or higher for collections loop!
Example – what does this do?

```java
int sum = 0;
String phrase = "3 weeks left";
for (char ch: phrase.toCharArray())
{
    sum = sum + 1;
}
```
Looping over a String – another way

• Use for loop instead – like complicated loop in Alice
• Like Alice, getting one item-from-list at a time
• Assume string variable is called words

for (int item=0; item< words.length(); item = item+1) 
{
    // do something here with words.charAt(item)
    // that is one character from words at a time
}
Conditionals – Format of “if”

- Must have ( )’s around condition!
- Can leave “else” part off

```java
if ( condition)
{
    // do if condition is true
}
else // can leave off if no else part
{
    // do if condition is false
}
```
Relational/Logic Operators

- Relational operators
  
  `<  >  <=  >=  ==  !=`

- Logic Operators
  
  - `&&` (and)
  - `||` (or)
  - `!` (not)

```java
if (((x > 0) && (y != 3)))
{
    // do something
}
```
Example – what does this do?

String letters = “CompSci 4 rocks”;
int sum = 0;
for (char ch: letters.toCharArray())
{
    if (ch == ‘S’ || ch == ‘s’)
    {
        sum = sum + 1;
    }
}
Problem 1 to Solve in Java

• Bioinformatics
  – Area of computer science
  – Application of computational techniques to the management and analysis of biological information

• Problem: Given a strand of DNA, determine the number of cytosine nucleotides present
Problem: Rewritten for CompSci

• DNA is a string – array of characters
  – Only has letters c, t, a and g
• Problem restated: how many c’s in a string?
• Example: “catacgtaagtc”
  – Answer: 3 c’s
• Write a method to return this number
  – See sheet for problem DNA-1
What does code mean?

• Name of class
• Name of method in class
• Return value (int is integer or number)
• One parameter (type and name)

```java
public class DNAProfile {
    public int count(String dna) {
        // fill in code here
    }
}
public class DNAProfile
{
    public int count(String dna)
    {
        // fill in code here
    }
}
How We Will Solve Problems in Java

• Write methods and test with testing interface: APT
  – Not a whole Java program, just a small part
• Write a complete Java program
  – Not yet

• Use a programming environment Eclipse to make it easier
• Use submission tool Ambient
• See CompSci 4 resources page to install!
Solve this Problem

- Write a method and test it on the APT
  - Type our solution into Eclipse
- Load the file into APT (web page) and test/run

Compsci 4, Spring 2009, APT

- Choose the problem you want to submit/test --- you should look at the problem statement, think about how to solve it, then write code to solve it. Next test your code via the online testing mechanism.
- Click Browse... to choose the file on your local system you'll test online.
- Click test/run to test the program.

If you use this page again you may have to reload/rebrowse for the program and reselect the problem.
Create a New Project in Eclipse

• Start Eclipse
• Select File -> New -> Project
  – Select Java Project and Next
  – Enter Project Name CPS4Sec1DNA
  – Think of project as an Alice world with lots of classes
Create a Class and Method

• Click on project CPS4Sec1DNA
  – Select File -> New -> Class
  – Enter name DNAprofile
  – Select Finish
  – DNAprofile window appears
  – Cut and paste the method “count” from the web page to the class
  – Complete the method

• Put all classes you create today in the same project!
Testing a method using APT

• Use APT to test method
• Select problem, load file, test/run.
• **Class laptops – file is in C: workspace**

Comp sci 4, Fall 2009, APT

- Choose the problem you want to submit/test --- you should look at the problem statement, think about how to solve it, then write code to solve it. After you've tested via the online testing mechanism you'll need to check it off in class.
- Click *Browse...* to choose the file on your local system you'll test online.
- Click *test/run* to test the program

If you use this page again you may have to reload/ rebrowse for the program and reselect the problem.

<table>
<thead>
<tr>
<th>Problem Set 1</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>For Classwork Nov 12</td>
<td></td>
</tr>
<tr>
<td>○ DNAprofile</td>
<td>DNA-1, Count C's, we will do together in class</td>
</tr>
<tr>
<td>○ DNAscgdiff</td>
<td>How Many More C's than G's</td>
</tr>
<tr>
<td>○ DNAcgcounr</td>
<td>DNA-2, CG Ratio</td>
</tr>
</tbody>
</table>

Test file: 

[Submit test/run]
Want Green, not red!

- Execution of the apt

**Problem:** dna1  
**Language:** java  
**Files:**  
- DNAprofile.java... DNAprofile ... java ... 20 line(s).

```
javac DNAprofile.java
```

Compilation successful.  
**Program running:** standard output below

```
Test Results Follow (scroll to see all)

# of correct: 3 out of 12

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>fail</td>
</tr>
<tr>
<td>2</td>
<td>pass</td>
</tr>
<tr>
<td>3</td>
<td>fail</td>
</tr>
<tr>
<td>4</td>
<td>fail</td>
</tr>
<tr>
<td>5</td>
<td>fail</td>
</tr>
<tr>
<td>6</td>
<td>fail</td>
</tr>
<tr>
<td>7</td>
<td>pass</td>
</tr>
</tbody>
</table>
Debugging your program

- Scroll down to see more detail
- Shows expected value, calculated value, and input value

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fail</td>
<td>pass</td>
<td>fail</td>
</tr>
<tr>
<td>expected</td>
<td>5</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>got</td>
<td>13</td>
<td>&quot;cgaatcgacgcgcg&quot;</td>
<td>&quot;g&quot;</td>
</tr>
<tr>
<td>got</td>
<td>&quot;cgaatcgacgcgcg&quot;</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>got</td>
<td>&quot;cgaatcgacgcgcg&quot;</td>
<td>&quot;cgaatcgacgcgcg&quot;</td>
<td>1</td>
</tr>
</tbody>
</table>
Saving your work to your Duke Account – if on class laptop

- Check in your project by selecting “Ambient”, “Check in project”
- First time only (Window -> preferences -> ambient -> checkin/checkout -> setup CVS)
- Enter your Duke account password
- If partner wants to save after one has saved, must click on project, select “Team”, then “disconnect”, then partner can try to save
Classwork today

• Solve the three APTs on the CompSci 4 APT web page *(create one Java project with three classes)*
  – DNA-1 CGTA counting
  – DNAcgdiff
  – DNA-2 CG counting
• Get work checked off – show runs and code
• If on class laptop, save files on Duke account
  – Ambient check in
    – **FIRST TIME only** *(window -> preferences -> ambient -> checkin/checkout - setup CVS repository)*