Announcements

• Assignment 7
  – Storyboard due Nov 27
  – Alice world due Nov 29

• More APT’s
  – String parts, Arrays
Review from last time

• What is Eclipse?
  – Environment to aid you in writing Java code
  – Create one project for each classwork
  – Put each new class in that same project

• What is an APT?
  – Tester for one method at a time

• Submit project (all classes for that day) with Ambient
Review Strings

- `String word = "CompSci 4";`
- `word.length()` – returns length of string
- `word.toCharArray()` – returns string as an array of characters
- `word.charAt(5)` – returns character at position 5
- Loop over characters in a string
  ```java
  for (char ch : word.toCharArray)
  {
  }
  ```
More on Strings

• `word.indexOf("Sci")`
  – Returns first position of “Sci” in word or –1 if not in word

• `word.substring(4,6);`
  – Returns string that starts at position 4, goes up to but not including position 6 and is of length 2

• `word.substring(4)`
  – Returns string that starts at position 4 til the end of the string

• `word = word + " rocks";`
  – Build a string – append to the right end
Print out a value

- System.out.println(String value);
- Prints out on one line
Example

String course = "CompSci 4 Spring 2006";
System.out.println(course);

int pos = course.indexOf("Spring");
String part1 = course.substring(0, pos);
String part2 = course.substring(pos+6);
course = part1 + "Fall" + part2;

System.out.println(course);
if – else if - else

- Can have as many “else if” as you want
- else is optional
- First case that is true is executed

- See example on next page, what happens when num=3? num=6? num=10?
What is output for values of num?

```java
if (num > 8)
{
    System.out.println(num);
}
else if (num > 5)
{
    System.out.println(num);
}
else
{
    System.out.println(num);
}
```
Arrays

• Parameter: double [] numbers
  – Means an array of doubles
  – Name of array is numbers
• Loop over items in an array – collections loop
  for (double num : numbers)
  {
    // do something with num
    // num is one item at a time from numbers
  }
Arrays – (loop if Java 1.4)

- Parameter: `double [] numbers`
  - Means an array of doubles
  - Name of array is `numbers`

- Loop over items in an array
  ```java
  for (int k = 0; k < numbers.length; k++) {
    // do something with numbers[k]
    // kth double in array numbers
  }
  ```
  - Note length applied to arrays – no parens
Array Example

• Find max number, assume at least one number in the array

```java
public class MaxInArrayTest {
    public double MaxInArray (double[] numbers) {
        double max = numbers[0];
        for (double num: numbers) {
            if (num > max) {
                max = num;
            }
        }
        return max;
    }
}
```
Array Example (loop if Java 1.4)

- Find max number, assume at least one number in the array

```java
public double MaxInArray(double[] numbers) {
    double max = numbers[0];
    for (int k = 1; k < numbers.length; k++) {
        if (numbers[k] > max) {
            max = numbers[k];
        }
    }
    return max;
}
```
Problem: DNA Max

- Given an array of DNA strands and a nucleotide (a, c, g, or t)
- Return the strand with the most occurrences of the nucleotide
- If there is more than one strand with the max number, return the longest such strand
Example

• Given array
  [“agt”, “aagt”, “taattt”, “ccatg”]
• Given nucleotide “a”
• Returns “taattt”
• “a” appears 2 times max in a strand
• Longest such strand is “taattt”
Solve this problem in Eclipse

```java
public class DNAMaxNucleotide {
    public String max(String[] strands, String n) {
        // fill in code here
    }
}
```

- What steps do we do?
- What do we already know how to do?
Classwork today – More APTs

• Test java methods using APT
• Create one new Java project called CPS4Sec1Nov20 for all classwork
• Create three new classes based on APT problems – test with APT
  – Class and methods must be spelled exactly as shown
• Submit the project for grading under today’s date
Class: **DNAComplement**
- Name of method: complement
- Build and return a new string with complements

Class: **DNAReverse**
- Name of method: reverse
- Build and return a new string that is the reverse

Class: **LongStrand**
- Name class: longest
- Return string with most nucleotides