Previous Lecture:
- Polymorphism

Today's Lecture:
- Object class
- Abstract
- char and String

Reading: Sec 4.5, 4.7
(Strings: Sec 5.2.1-5.2.3)

---

Accessing methods/variables through polymorphic references

Reference type ≠ object type:
1. What determines whether a method/variable can be accessed?
2. For an overridden method, what determines which version gets invoked?

---

The Object class

If a class is not explicitly defined to be the child of an existing class, it is assumed to be the child of the Object class
⇒ All classes are derived from the Object class

```java
class Room
    is the same as

class Room extends Object
```

---

abstract class

- A placeholder in a class hierarchy that represents a generic concept
- Cannot be instantiated
- Modifier: abstract

```java
public abstract class Geometry
```

- Can contain abstract methods
  ```java
  public abstract double Area();
  ```
- Subclasses of abstract classes will “fill out” these abstract methods

---

Example: password

Write a code fragment to generate a password according to the following rules:

- 8 characters: 1st 6 letters or digits, last 2 digits
- Each of first 6: equally likely to be letter or digit
  - If letter, equally likely to be any letter (A-Z)
  - If digit, equally likely to be any digit (0-9)
- Last 2 digits: sum of digits in first 6 positions
ASCII characters

<table>
<thead>
<tr>
<th>ascii code</th>
<th>Character</th>
<th>ascii code</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>65</td>
<td>'A'</td>
<td>48</td>
<td>'0'</td>
</tr>
<tr>
<td>66</td>
<td>'B'</td>
<td>49</td>
<td>'1'</td>
</tr>
<tr>
<td>67</td>
<td>'C'</td>
<td>50</td>
<td>'2'</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>90</td>
<td>'Z'</td>
<td>57</td>
<td>'9'</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

What we learned...

- Develop/implement algorithms for problems
- Develop programming skills
  - Design, test, debug, document, demonstrate
- Apply programming languages
  - Control structures
  - Function/methods for reducing redundancy
  - Data structure
  - Fundamentals of object oriented programming, including inheritance

What we learned... (cont’d)

- Specific tasks
  - Sorting
  - Simulation of systems
  - Text and string processing
  - Handling input/output
  - Plotting numeric data

Final Exam

- Tues, Dec 14, 9-11:30am, Uris Aud
- 2/3 Java, 1/3 MATLAB
- Closed-book exam, no calculators
- Bring student ID card

- Check for announcements on webpage:
  - Study break office/consulting hours
  - Review session time and location
  - Review questions
  - List of potentially useful functions/methods
public class CharNString{

    public static void main(String[] args) {

        int num= 8;               //length of password
        char[] c= new char[num];  //characters in password
        int sum= 0;               //sum of digits in password

        //Generate first 6 characters
        for (int i=0; i<num-2; i++)
            if (Math.random()<0.5)
                //generate character
            else {
                //generate digit
                int r=

                sum += r;
            }

        //"Calculate" last two digits

        System.out.println(c);

        //Make a String from a char array
        String word= new String(c);
        System.out.println("Length of word: " + word.length());

        //Make a char array from a String
        char[] letters= word.toCharArray();
        System.out.println("Length of array: " + letters.length());
    }
}