Last Time

• Design of three classes: Human, Computer, and Game
• Implementation of the Human class
• Implementation of the Computer and Game classes
• Implementation of the Player class, which we set as a superclass of the Human and CPU classes
Office Hours

<table>
<thead>
<tr>
<th>GMT-04</th>
<th>Mon 9/14</th>
<th>Tue 9/15</th>
<th>Wed 9/16</th>
<th>Thu 9/17</th>
<th>Fri 9/18</th>
</tr>
</thead>
<tbody>
<tr>
<td>8am</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9am</td>
<td>9 – 9:50 Astro</td>
<td>9 – 9:50 Astro</td>
<td>9 – 4p Research/Medical</td>
<td>9 – 9:50 Astro</td>
<td></td>
</tr>
<tr>
<td>10am</td>
<td>10 – 11 Office Hours (15min slots)</td>
<td>10 – 11 Office Hours (15min slots)</td>
<td>10 – 11 CS Department Meeting</td>
<td>10 – 11 CS Department Meeting</td>
<td></td>
</tr>
<tr>
<td>11am</td>
<td>11 – 11:50 CMPSC112</td>
<td>11:30 – 12:30p Lunch</td>
<td>11 – 11:50 CMPSC112</td>
<td>11 – 11:50 CMPSC112</td>
<td></td>
</tr>
<tr>
<td>12pm</td>
<td>12p – 1p Lunch</td>
<td>12:30p – 2p InfoVis@VT Web Meeting</td>
<td>12p – 1p Lunch</td>
<td>12p – 1p Lunch</td>
<td></td>
</tr>
<tr>
<td>1pm</td>
<td>1:30p – 2:20p CMPSC382</td>
<td>1:30p – 2:3 CMPSC600</td>
<td>1:30p – 2:20p CMPSC382</td>
<td>1:30p – 2:20p CMPSC382</td>
<td></td>
</tr>
<tr>
<td>2pm</td>
<td>2:30p – 3:30p – Office Hours (15min slots)</td>
<td>2:30p – 4:20p CMPSC382 Lab</td>
<td>2:30p – 4:20p CMPSC112 Lab</td>
<td>2:30p – 4:20p CMPSC382</td>
<td></td>
</tr>
<tr>
<td>3pm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2:30p – 5p Office Hours (15min slots)</td>
</tr>
<tr>
<td>4pm</td>
<td></td>
<td></td>
<td></td>
<td>4p – 5p VT Research Meeting</td>
<td></td>
</tr>
<tr>
<td>5pm</td>
<td></td>
<td></td>
<td>5p – 6p Faculty Meeting</td>
<td>4:20p – 5:5 Faculty Meeting</td>
<td>5p – 6p Astronomy</td>
</tr>
<tr>
<td>6pm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Constructors

• What is a constructor?
  – A special method used to initialize a newly created instance of a class.
  – Each class has an implicit default constructor, but we can add a custom one.

• Constructors are **not** inherited in Java.
  – You can access the constructor of the superclass with the keyword `super`.
  – If your custom constructor doesn’t make a call to `super()` as the first command, then an implicit call is made to the superclass default constructor.
Polymorphism

• A reference variable can take on different forms, either of the declared class or of any related subclass:

    Player human, cpu;
    human = new Human();
    cpu = new CPU();

• Unfortunately, the human variable can’t access any methods exclusive to the Human class.
  – Solution: Abstract classes
Abstract

• An **abstract class** contains some methods that are declared but not implemented.
  – Methods that are not implemented are **abstract methods**.
  – Methods that are implemented are **concrete methods**.

• By making the Player class abstract and adding the missing method declarations, we can now make our Human and CPU variables **polymorphic**.
Dynamic Dispatch

• So now we have a variable `computerPlayer`, which is declared as type `Player` and initialized as type `Computer`. Which `increaseScore()` function gets called?
  
  – Java decides at runtime to call the version of the method that is most specific to the initialized type rather than the declared type.
  
  – (Our CPU is still cheating with double points.)
• Need to check whether the Player object you created is a Human or a CPU?

```java
if (var1 instanceof Human) {
    // do stuff
} else if (var1 instanceof CPU) {
    // do other stuff
} //if-else
```
Any Questions?