

## CSE 142 Final Cheat Sheet

Math Methods:	Description
<code>Math.abs (value)</code>	absolute value
<code>Math.min (v1, v2)</code>	smaller of two values
<code>Math.max (v1, v2)</code>	larger of two values
<code>Math.round (value)</code>	nearest whole number
<code>Math.pow (b, e)</code>	b to the e power

Random Method	Description
<code>nextInt (max)</code>	random integer from 0 to $max-1$

### Construction Examples

```
int[] data = new int[10];
Random r = new Random();
```

**Types:** `ArrayList<String>`, `ArrayList<Integer>`

String Methods:	Description
<code>contains (str)</code>	true if this string contains the other's characters inside it
<code>endsWith (str)</code> , <code>startsWith (str)</code>	true if this string starts/ends with the other's characters
<code>equals (str)</code>	true if this string is the same as <code>str</code>
<code>equalsIgnoreCase (str)</code>	true if this string is the same as <code>str</code> , ignoring capitalization
<code>indexOf (str)</code>	index in this string where given string begins (-1 if not found)
<code>length ()</code>	number of characters in this string
<code>substring (i, j)</code>	characters in this string from index <code>i</code> (inclusive) to <code>j</code> (exclusive)
<code>toLowerCase ()</code> , <code>toUpperCase ()</code>	a new string with all lowercase or uppercase letters
<code>charAt (i)</code>	returns char at index <code>i</code>

Scanner Methods:	Description
<code>nextInt ()</code> , <code>hasNextInt ()</code>	read/return token as <code>int</code> and test if reading will succeed
<code>next ()</code> , <code>hasNext ()</code>	read/return token as <code>String</code> and test if reading will succeed
<code>nextDouble ()</code> , <code>hasNextDouble ()</code>	read/return token as <code>double</code> and test if reading will succeed
<code>nextLine ()</code> , <code>hasNextLine ()</code>	read/return line as <code>String</code> and test if reading will succeed

ArrayList Methods:	Description
<code>add (value)</code>	appends value at end of list
<code>add (index, value)</code>	inserts given value at given index, shifting subsequent values right
<code>clear ()</code>	removes all elements of the list
<code>get (index)</code>	returns the value at given index
<code>remove (index)</code>	removes/returns value at given index, shifting subsequent values left
<code>set (index, value)</code>	sets a value in the list at the given index; no other elements affected
<code>size ()</code>	returns the number of elements in the list

## Critter class template

```
public class name extends Critter {  
    fields  
  
    constructor  
  
    public boolean eat() {  
        statement(s) that return true (eat) or false (don't eat)  
    }  
  
    public Attack fight(String opponent) {  
        statement(s) that return either Attack.ROAR, Attack.POUNCE, or Attack.SCRATCH  
    }  
  
    public Color getColor() {  
        statement(s) that return a Color  
    }  
  
    public Direction getMove() {  
        statement(s) that return either  
        Direction.NORTH,  
        Direction.SOUTH,  
        Direction.EAST,  
        Direction.WEST, or  
        Direction.CENTER  
    }  
  
    public String toString() {  
        statement(s) that return a String  
    }  
}
```