

CSE 142 Final Cheat Sheet

Syntax templates:

Declaring and using arrays

```
type[] name = new type[length];  
type[] name = {VAL1, VAL2, VAL3, ...};  
name[index] = value;  
name.length // number of elements in array
```

Declaring objects

```
Type objectName = new Type(parameters);
```

Classes

Field (data inside each object)

```
private type name;
```

Method (behavior inside each object)

```
public type name(parameters) {  
    statement(s);  
}
```

Constructor (code to initialize new objects)

```
public className(parameters) {  
    statement(s);  
}
```

toString method (called when an object is printed)

```
public String toString() {  
    code that produces/returns a String;  
}
```

Inheritance

```
public class name extends superclass {  
    field(s), constructor(s), method(s), etc.  
}
```

Critter class template:

```
public class name extends Critter {
```

field(s)

constructor(s)

```
public Color getColor() {  
    statement(s) that return a Color (e.g. Color.RED, Color.BLUE, Color.WHITE)  
}
```

```
public Action getMove(CritterInfo info) {  
    statement(s) that return Action.INFECT, Action.HOP, Action.LEFT, OR Action.RIGHT  
}
```

```
public String toString() {  
    statement(s) that return a String  
}
```

```
}
```

CritterInfo Method	Description
getFront(), getBack(), getLeft(), getRight()	returns one of Neighbor.WALL, Neighbor.EMPTY, Neighbor.SAME, Neighbor.OTHER
getDirection()	returns one of Direction.NORTH, Direction.SOUTH, Direction.EAST, Direction.WEST

Random Method	Description
nextInt(<i>max</i>)	returns a random integer from 0 to <i>max</i> -1

Math Method	Description
Math.abs(<i>value</i>)	returns the absolute value
Math.min(<i>v1</i> , <i>v2</i>)	returns the smaller of two values
Math.max(<i>v1</i> , <i>v2</i>)	returns the larger of two values
Math.round(<i>value</i>)	returns the nearest whole number
Math.sqrt(<i>value</i>)	returns the square root
Math.pow(<i>base</i> , <i>exp</i>)	returns base to the exponent power

String Method	Description
contains(<i>str</i>)	returns <code>true</code> if this string contains the other's characters inside it
endsWith(<i>str</i>), startsWith(<i>str</i>)	returns <code>true</code> if this string starts/ends with the other's characters
equals(<i>str</i>)	returns <code>true</code> if this string is the same as <i>str</i>
equalsIgnoreCase(<i>str</i>)	returns <code>true</code> if this string is the same as <i>str</i> , ignoring capitalization
indexOf(<i>str</i>)	returns the first index in this string where given string begins (-1 if not found)
length()	returns the number of characters in this string
replace(<i>str1</i> , <i>str2</i>)	replace all occurrences in this string of <i>str1</i> with <i>str2</i>
substring(<i>i</i> , <i>j</i>)	returns characters in this string from index <i>i</i> (inclusive) to <i>j</i> (exclusive)
substring(<i>i</i>)	returns characters in this string from index <i>i</i> to end (inclusive)
toLowerCase(), toUpperCase()	returns a new string with all lowercase or uppercase letters
charAt(<i>i</i>)	returns char at index <i>i</i>

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

ArrayList Method	Description
add(<i>value</i>)	appends given value at end of list
add(<i>index</i> , <i>value</i>)	inserts given value at given index, shifting subsequent elements right
get(<i>index</i>)	returns value at given index
remove(<i>index</i>)	removes/returns value at given index, shifting subsequent elements left
set(<i>index</i> , <i>value</i>)	replaces value at given index with given value
size()	returns number of elements in list