

CSE 142 Final Cheat Sheet

```

for (initialization; test; update) {
    statement(s);
    ...
}

if (test) {
    statement(s);
} else if (test) {
    statement(s);
} else {
    statement(s);
}

type name = value; // variable declaration and initialization

Type objectName = new Type(parameters); // object construction

```

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

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

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

Scanner Method	Description
nextInt()	reads/returns input token as int
next()	reads/returns input token as String
nextDouble()	reads/returns input token as double
nextLine()	reads/returns line as String
hasNextInt()	returns true if there is a next token and it can be read as an int
hasNext()	returns true if there is a next token to read
hasNextDouble()	returns true if there is a next token and it can be read as a double
hasNextLine()	returns true if there is a next line to read

Declaring and using Arrays

```
type[] name = new type[length];  
name[index] = value;
```

Arrays Class Method	Description
<code>Arrays.fill(array, value)</code>	sets every element in the array to the given value
<code>Arrays.sort(array)</code>	arranges the elements in the array into ascending order
<code>Arrays.toString(array)</code>	returns a <code>String</code> for the array, such as: "[10, 30, 17]"
<code>Arrays.copyOf(array, length)</code>	returns a new copy of the given array of the given length
<code>Arrays.equals(array1, array2)</code>	returns <code>true</code> if the two arrays have the same elements

Classes

Field (data inside each object)

```
private type name;
```

Method (behavior inside each object)

```
public type name(parameters) {  
    statements;  
}
```

Constructor (code to initialize new objects)

```
public className(parameters) {  
    statements;  
}
```

toString method (called when an object is printed)

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

Inheritance

```
public class name extends superclass {  
}
```

Critter classes

```
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) {  
        returns 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;  
    }  
}
```

```
System.out.printf("format string", value, ..., value);
```