

CS 142 FINAL EXAM CHEAT SHEET

```

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

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

while (condition) {
    statement(s);
}

public static type name(parameters) {
    statement(s);
    ...
    return expression;
}
    
```

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(max)	random integer from 0 to <i>max</i> -1

String Method	Description
contains(str)	true if this string contains the other's characters inside it
endsWith(str), startsWith(str)	true if this string starts/ends with the other's characters
equals(str)	true if this string is the same as <i>str</i>
equalsIgnoreCase(str)	true if this string is the same as <i>str</i> , ignoring capitalization
indexOf(str)	index in this string where given string begins (-1 if not found)
length()	number of characters in this string
replace(str1 , str2)	replace all occurrences in this string of <i>str1</i> with <i>str2</i>
substring(i , j)	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)	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

<code>nextLine()</code>	reads/returns line as <code>String</code>
<code>hasNextInt()</code>	returns <code>true</code> if there is a next token and it can be read as an <code>int</code>
<code>hasNext()</code>	returns <code>true</code> if there is a next token to read
<code>hasNextDouble()</code>	returns <code>true</code> if there is a next token and it can be read as a <code>double</code>
<code>hasNextLine()</code>	returns <code>true</code> if there is a next line to read

Declaring and using Arrays

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

ArrayList

```
ArrayList<Integer> list = new ArrayList<Integer>();
```

<code>add(value)</code>	adds value to collection (appends 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 collection
<code>contains(value)</code>	returns <code>true</code> if the given value is found somewhere in this collection
<code>equals(collection)</code>	returns <code>true</code> if the given other collection contains the same elements
<code>get(index)</code>	returns the value at given index
<code>indexOf(value)</code>	returns first index where given value is found in list (-1 if not found)
<code>isEmpty()</code>	returns <code>true</code> if the collection has no elements
<code>remove(value)</code>	finds and removes the given value from this collection
<code>set(index, value)</code>	replaces value at given index with given value
<code>size()</code>	returns the number of elements in the collection
<code>toString()</code>	returns a string representation such as "[10, -2, 43]"

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 {}
```

Interface

```
public class name implements interface {}
```