CSE 143X Midterm Cheat Sheet

Syntax templates

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

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

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

while (condition) {
    statement(s);
}
```

```
public static void name(parameters) {
    statement(s);
}

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

for (int i = 0; i < array.length; i++) {
    do something with array[i];
    ...
}

for (int i = 0; i < string.length(); i++)
{
    do something with string.charAt(i);
    ...
}
```

Math Method	Description
`Math.abs(`*value*`)`	absolute value
`Math.min(`*v1, v2*`)`	smaller of two values
`Math.max(`*v1, v2*`)`	larger of two values
`Math.round(`*value*`)`	nearest whole number
`Math.pow(`*b, e*`)`	b to the e power

Random Method	Description
`nextInt(`*max*`)`	random integer from 0 to *max*-1

Construction Examples

```
int[] data = new int[10];
Random r = new Random();
Scanner console = new Scanner(System.in);
```

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 *str*
`equalsIgnoreCase(`***str***`)`	true if this string is the same as *str*, ignoring capitalization
`indexOf(`***str***`)`	index in this string where given string begins (-1 if not found)
`length()`	number of characters in this string
`substring(`***i***`, `***j***`)`	characters in this string from index *i* (inclusive) to *j* (exclusive)
`toLowerCase()`, `toUpperCase()`	a new string with all lowercase or uppercase letters
`charAt(`***i***`)`	returns char at index *i*

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

Operator	Description
<	less than
<=	less than or equal
>	greater than
>=	greater or equal
==	equal
!=	not equal

Operator	Description
&&	and
||	or
!	not