CS 210 Midterm 1 Review

Starting out

```
import java.util.Scanner;
import java.util.Random;
import java.util.Arrays;
import java.util.ArrayList;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintStream;


public class ClassName{
    //Static vars
    static int ROWS = 10;

    public static void main(String[] args) throws FileNotFoundException{
       double d = method(10);
    }

    public static double method(int n){
        return n;
    }

}
```

Type Casting

```
double result = (double) 19 / 5; // 3.8
int result2 = (int) result; // 3
int x = (int) Math.pow(10, 3); // 1000

String a="12.3";
double d = Double.valueOf(a); // String to double
System.out.println(d+1); // 13.3

String b="1";
int i = Integer.valueOf(b); // String to int
System.out.println(i+1); // 2
```

Methods

```
// public or private
// static (accessible)
// return type
// MUST ALWAYS HAVE RETURN OUTPUT FOR ALL PATHS
public static int adder(int a, int b){
    return a+b;
}
```

For Loops

```
// Starts at 0 and ends at 5
// 0 1 2 3 4 5 (loops 6 times)
for(int i = 0; i<=5; i++) {
    for(int j = 0; j<=5; j++) {
    }
}

// 0  1 2 3 4  (loops 5 times)
for(int i = 0; i<5; i++) {
}

// 1 2 3 4 5  (loops 5 times)
for(int i = 1; i<=5; i++) {
}
```

While and Do While Loops

```
// checks condition at the start
while (condition) {

}

// checks condition at the end
do {

} while (condition);
```

if statement

```
//Ternary
(condition) ? true_statement : false_statement;

if(condition){

}
// !condition (not)
// condition1 && condition2
// condition2 || condition2
```

Specifics

Strings

```
// String var_name = "StringName";
String s = "Hello World"

s.charAt(0) // H
```

Methods	Return
`String.length()`	`int`
`String.equals(str)`	`boolean`
`String.indexOf(str)`	`int`
`String.substring(index1, index2)`	`String` index2 defaults to `string.length()-1`
`String.toLowerCase()`	`String`
`String.toUpperCase()`	`String`
`String.charAt(Index)`	`char`
`String.equalsIgnoreCase(str)`	`boolean`
`String.startsWith(str)`	`boolean`
`String.endsWith(str)`	`boolean`
`String.contains(str)`	`boolean`

System.out.printf

```
//default right align
System.out.printf(string, params);
System.out.printf("my teacher is %s", "taesik kim");
```

Methods	Return
%n	Line Break
%b	boolean
%c	char
%s	String
%d	int
%f	float
%%	% symbol
%-6.1f	example left align 6 chars wide 1 decimal.

Chars

```
// A - 65 / Z - 90
// a - 97 / z - 122
char c1 = 'b';
char c2 = 'd';

int n = c1 + c2; // 98 + 100 = 198
boolean test = (c1 == c2); // false

for(char c ='a'; c<='f'; c++){
    System.out.println(c + " " + (1+c));
    // a 98
    // b 99
}
```

Random

```
Random r = new Random();

r.nextInt(10); // (0 - 9) inclusive

public static int randomIntRange(int min, int max) {
    return new Random().nextInt(max - min + 1) + min; //range = (max - min + 1)
}
```

Methods	Return
`r.nextInt(max)`	`int` [0, max)
`r.nextDouble()`	`double` [0.0, 1.0)

Scanners

```
import java.util.Scanner;

//Scanner var = new Scanner(input type);
// input type: System.in, new File(), String s;
Scanner s = new Scanner(System.in);
int n = s.nextInt();

s.close();
```

Methods	Return
`s.next()`	`String`
`s.nextLine()`	`String`
`s.nextInt()`	`int`
`s.nextDouble()`	`double`
`s.hasNext()`	`boolean`
`s.hasNextInt()`	`boolean`
`s.hasNextDouble()`	`boolean` (can take int)

File Input/Output

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintStream;

// main method stuff
public static void main(String[] args) throws FileNotFoundException {
    // file location
    File fin = new File("member.txt");
    File fout = new File("output.txt");

    Scanner fileinput = new Scanner(fin);
    PrintStream out = new PrintStream(fout);

    String readinput = fileinput.nextLine();
    out.println(readinput);

    fileinput.close();
    out.close();
}
```

Methods	Return
`PrintStream.close()`	`void`
`PrintStream.println(output)`	`void`

Arrays

```
// ArrayType[] var_name = new ArrayType[size];
// Static size

int[] numbers = new int[8];
String[] s1 = {"hi", "bye", "good night"};
String[] s2 = s1;
// NOT copied. referenced so s1 and s2 are the same object array.

numbers[0] = 100;
numbers[7] = 90;

for(int n : numbers) {

}
```

Methods	Return
`array.length`	int
`array[index]`	arrayType

> Import java.util.Arrays

Methods	Return
`Arrays.binarySearch(array,value)`	
`Arrays.copyOf(array, length)`	
`Arrays.equals(array1, array2)`	boolean (same order)
`Array.fill(array, value)`	array
`Arrays.sort(array)`	array
`Arrays.toString(array)`	string