

AP CS A EXAM REVIEW SHEET (NICK WALD)

WRITE DOWN WHAT THE QUESTION IS ASKING!

CHECKLIST

- ☐ Do I know what the question is asking?
- ☐ What is the return type?
- ☐ Do I need to make a variable?
- ☐ Do I need a loop?
- ☐ What needs to be in the If statement?
- ☐ What is returned?

FOR OPEN-ENDED QUESTIONS:

- ☐ MAKE VARIABLE?
 - ☐ IF SO, PRIVATE? PUBLIC?
- ☐ MAKE SETTER OR GETTER?
- ☐ WHAT HAPPENS TO CONSTRUCTOR?

FOR ARRAYLIST QUESTIONS

- ☐ Make a variable
- ☐ Set that variable somehow (either starting value using = is static), using a new constructor with a new argument or making a setter
- ☐ Make sure to write a getter
- ☐ Maybe reference how/where that variable will change or how it applies to the problem

Iterative Statements

```
// for loop
for (condition){
    expression;
}
// for each loop
for (int i: someArray){
    expression;
}
// while loop
while (condition){
    expression;
}
```

Variables

```
{public|private} [static] type name [= expression|value];
```

Methods

```
{public|private} [static] {type | void} name(arg1, ..., argN ){
    statements
}
```

Decisive Methods

```
//if statement
if(condition){
    expression
}
//if-else statement
if(condition){
    expression
}
else{
    expression
}
```

String Methods

```
// Creating String using literal
String str1 = "Welcome";
// Creating String using new keyword
String str2 = new String("Edureka");

str1 == str2 //compare the address;
String newStr = str1.equals(str2); //compares the values
String newStr = str1.equalsIgnoreCase();
newStr = str1.length(); //calculates length
newStr = str1.charAt(i); //extract i'th character
newStr = str1.toUpperCase(); //returns string in ALL CAPS
newStr = str1.toLowerCase(); //returns string in ALL LOWERCASE
newStr = str1.replace(oldVal, newVal); //search and replace
newStr = str1.trim(); //trims surrounding whitespace
newStr = str1.contains("value"); //Check for the values
newStr = str1.toCharArray(); //Convert into character array
newStr = str1.isEmpty(); //Check for empty String
newStr = str1.endsWith(); //Checks if string ends with the given suffix
```

Escape Sequences

```
\t
tab
\n newline
\\ backslash
```

Arrays

use .length

Declaration:

```
//dataType[] name;
double[] apList;
```

Instantiation:

```
//dataType[] listName = new dataType[arraySize]
double[] apList = new double[10];
```

Common methods:

```
//loop thru array and print each item:
public static void printArray(int[] array) {
    for (int i = 0; i < array.length; i++) {
        System.out.print(array[i] + " ");
    }
}
```

```
//reversing an array:
for(int i=0; i<(arr.length())/2; i++){
    double temp = a[i];
    a[i] = a[n-1-i];
    a[n-1-i] = temp;
}
```

ArrayList

use .size();

Methods:

ADD

```
listName.add(value);
//adds parameter to end of arrayList
listName.add(index, value);
//adds value at specified index, shifts all other indexes up by 1
```

```
arrList.add("hi");
//final value in arrList is now hi
```

```
arrList.add(0, "nick");
//value at position 0 becomes nick
//value @ pos 0 shifts to pos 1
//value @ pos 1 shifts to pos 2, etc...
```

SET

```
listName.set(index, value)
//whatever is in position index has now been changed to the new value,
```

```
arrList.set(0, "owen")
//the value at position 0 becomes owen
```

Replaces value at specified position with new specified value

GET

```
arrList.get(0);
//returns value at position 0
```

Return value at specified position

REMOVE

```
arrList.remove(1);
```

Removes value at specified index, shifts all subsequent indexes down by 1

REQUIRES AN `i--` AFTER IN A LOOP

QUICK REFERENCE

class java.lang.Object

- boolean .equals(Object other)

- `String .toString()`

class java.lang.Integer

- `Integer(int value)`
- `int intValue()`
- `Integer.MIN_VALUE`
minimum value represented by an int or Integer
- `Integer.MAX_VALUE`
maximum value represented by an int or Integer

class java.lang.String

- `int .length()`
- `String .substring(int from, int to)`
returns the substring beginning at from and ending at to-1
- `String substring(int from)`
returns `substring(from, length())`
- `int .indexOf(String str)`
returns the index of the first occurrence of str;
returns -1 if not found
- `int .compareTo(String other)`
returns a value < 0 if this is less than other
returns a value = 0 if this is equal to other
returns a value > 0 if this is greater than other

class java.lang.String

- `int length()`
- `String substring(int from, int to)`
returns the substring beginning at from and ending at to-1
- `String substring(int from)`
returns `substring(from, length())`
- `int indexOf(String str)`
returns the index of the first occurrence of str;
returns -1 if not found
- `int compareTo(String other)`
returns a value < 0 if this is less than other
returns a value = 0 if this is equal to other
returns a value > 0 if this is greater than other

interface java.util.List

- `int size()`
- `boolean add(E obj)`
appends obj to end of list; returns true
- `void add(int index, E obj)`
inserts obj at position index <= 0 <= size; moving elements at position index and higher to the right (adds 1 to their indices) and
- `E get(int index)`
- `E set(int index, E obj)`
replaces the element at position index with obj
returns the element formerly at the specified position
- `E remove(int index)`
removes element from position index, moving elements at position index + 1 and higher to the left (subtracts 1 from their indices)
returns the element formerly at the specified position