

Back-pocket Code - CS60 - Fall 2016

The goal is for you to have some code in Java and Racket in your back-pocket. To do this, you'll first understand and memorize this code. Once you have it memorized, you'll be able to pull this code out (metaphorically) to help you solve more complex problems. And then you'll be able to focus on that new complexity and not the syntax or basic structures you need to solve that problem.

This is a new teaching strategy that we're trying this semester, but education/psychology research regarding [cognitive load](#), [chunking](#), [worked examples](#), and [the importance of domain knowledge](#) suggests this is a reasonable idea. Imagine trying to solve Algebra problems if you had to spend a lot of time thinking about addition, subtraction, multiplication, and division. We hope that this code becomes automatic like those math skills and then allows you to solve more interesting problems!

The first quiz requires that you memorize the code. If you get less than 95% on the quiz, you'll need to take the quiz again (and again) until you get above 95%. You won't be graded on: variable names

We've attached 7 blank copies of the quiz so that you can practice! There is a link to this document on Piazza, and it includes links to a video for all of the back-pocket code! Watch those videos :-)

Array Access (Java) - [video](#)

Write a Java method named `swap` that swaps the elements in two indices of an array holding `ints`.

Learning goals: swap algorithm and syntax for: method and variable declaration, array access & assignment.

```
public static void swap(int index1, int index2, int[] myArray) {  
    int temp = myArray[index1];  
    myArray[index1] = myArray[index2];  
    myArray[index2] = temp;  
}
```

For Each List Element (Racket) - [video](#)

Write a Racket function named `incrementAll` that adds 1 to each of the numbers in an input List.

Learning goals: Racket conditionals, linear recursion, '()', List functions: `null?`, `cons`, `first`, `rest`

```
(define (incrementAll myList)  
  (if (null? myList)  
      '()  
      (cons (+ 1 (first myList))  
            (incrementAll (rest myList))))))
```

For Each Array Element (Java) - [video](#)

Write a Java method named `incrementAll` that adds 1 to each of the numbers in an input `int` array.

Learning goals: Java for-loop syntax, ++ operator

```
public static void incrementAll(int[] myArray) {  
    for (int index = 0; index < myArray.length; index++) {  
        myArray[index]++;  
    }  
}
```

Filter an ArrayList (Java) - [video](#)

Write a Java method named `keepShortWords` that returns a new `ArrayList` with only the input words with less than 5 characters.

Learning goals: Java conditionals, ArrayList construction and methods: size, get, & add. String method: length

```
public static ArrayList<String> keepShortWords(ArrayList<String> inputList) {  
    ArrayList<String> outputList = new ArrayList<String>();  
    for (int index = 0; index < inputList.size(); index++) {  
        String currentWord = inputList.get(index);  
        if (currentWord.length() < 5) {  
            outputList.add(currentWord);  
        }  
    }  
    return outputList;  
}
```

Filter a List (Racket) - [video](#)

Write a Racket function that returns a list with only the even numbers from the input list.

Learning goals: Racket `cond` and `modulo`

```
(define (keepEvenNumbers myList)  
  (cond [(null? myList)  
        '()]  
        [(= 1 (modulo (first myList) 2))  
         (keepEvenNumbers (rest myList))]  
        [else  
         (cons (first myList)  
               (keepEvenNumbers (rest myList)))])  
  ))
```

Practice Exam - Back-pocket Code

CS60 - Fall 2016

We recommend you answer the questions in a different order each time to make it a little harder:-)

Array Access (Java)

Write a Java method named `swap` that swaps the elements in two indices of an array holding `ints`.

For Each List Element (Racket)

Write a Racket function named `incrementAll` that adds 1 to each of the numbers in an input List.

For Each Array Element (Java)

Write a Java method named `incrementAll` that adds 1 to each of the numbers in an input `int` array.

Filter an ArrayList (Java)

Write a Java method named `keepShortWords` that returns a new `ArrayList` with only the input words with less than 5 characters.

Filter a List (Racket)

Write a Racket function that returns a list with only the even numbers from the input list.

Back-pocket Code Exam

CS60 - Fall 2016 - September 12, 2016

Please wait to get started until Bryan Mehall says to start.
You can sign this honor code statement before we start.

I will not use any resources other than my knowledge of the topic to complete this exam (i.e. I won't reference the correct answer) and I will not look at any other student's solution as I complete my exam.

Name: _____

Signature: _____

	Total
Array Access (Java)	/10
For Each List Element (Racket)	/10
For Each Array Element (Java)	/10
Filter an ArrayList (Java)	/10
Filter a List (Racket)	/10

Array Access (Java)

Write a Java method named `swap` that swaps the elements in two indices of an array holding `ints`.

For Each List Element (Racket)

Write a Racket function named `incrementAll` that adds 1 to each of the numbers in an input `List`.

For Each Array Element (Java)

Write a Java method named `incrementAll` that adds 1 to each of the numbers in an input `int` array.

Filter an ArrayList (Java)

Write a Java method named `keepShortWords` that returns a new `ArrayList` with only the input words with less than 5 characters.

Filter a List (Racket)

Write a Racket function that returns a list with only the even numbers from the input list.