

## Programming HW 7

CS 169 Spring 19, Linnell

Scala

Due Friday 5/31 in class

Submission instructions at the end

1. (5 points) Use Call by Name to write a function `forLoop` in Scala that mimics the behavior of a typical for loop from procedural programming. `forLoop` should have two parameter groups. The first parameter group should take in an integer value `i`, a condition function that must yield true when applied to `i` in order for the loop to continue, and an increment function that when applied to `i` yields the value of `i` on the next iteration of the loop. The second parameter group should contain a single parameter of type `Unit`, the expression to be evaluated each time through the loop. Write a function call to `forLoop` that will print out "Hello World" 10 times. 1 point extra credit: What is the most important shortcoming of the function you've written when compared to a for loop in C++?
2. (15 points) This problem will use the notion of representing a set using its characteristic function that we discussed in class primarily on Wednesday 5/22 (all the code is posted there). You will build on the `contains`, `union`, `intersect`, and `diff` functions that we defined in class.
  - a. Define a function `def setList(List[Int]): Int=>Boolean` which takes in a list of Ints and returns a set consisting of all Ints in the list.
  - b. Define the function `def filter(s: Int=>Boolean, p: Int=>Boolean): Int=>Boolean` which takes in a set `s` and a predicate `p`, and returns a new set that consists of the elements of `s` that satisfy the predicate. You MUST use an anonymous function.
  - c. Define a function `def forall(s: Int=>Boolean, p: Int=>Boolean): Boolean` that tests whether a predicate `p` is true for all elements of a set `s`. In order to make it possible to implement this function, we will consider a predicate true for all integers if it is true for integers from -1000 to 1000.
3. Classes (5 points). Fill in the class and functions below as described in the comments.

```
class Set(f: Int=>Boolean){
//the function f returns true for elements of the set and false for all other
numbers
  def contains(elem: Int): Boolean =
    //returns true if elem is in the set and false for all other numbers

  def \/ (t: Set): Set =
    //Returns the union of this set and t.

  def /\ (t: Set): Set =
    //Returns the intersection of this set and t
```

```

def - (t:Set):Set =
  //Returns the difference of this set and t

def filter(p:Int=>Boolean):Set =
  // Returns a new set that consists of the elements of s that satisfy the
  predicate.

def forall(p:Int=>Boolean):Boolean =
  //Returns true if the predicate is true for all elements of this set, and
  false otherwise.
  // In order to make it possible to implement this function, we will consider a
  predicate true for all integers if it is true for integers from -1000 to 1000.
}

```

Submission instructions: You will print out your code for each problem, stapling together multiple sheets (there will be deductions for unstapled homework!). Turn in the hardcopy at the beginning of class. You will ALSO submit your .scala files as an attachment, to cs169@math.scu.edu (NOT Dr. Linnell's email!!) The subject line of the email should be "CS169 HW8 YourLastName YourIDNumber "