CS 113 – Computer Science I

Lecture 09 – Recursion, Strings, Arrays

Tuesday 10/03/2023
Announcements

• HW03 – moved deadline to tonight 10/03

• HW04 – due Monday 10/09
  First write method stubs, and upload programs with method stubs to Gradescope
  Ensures method signatures are correct

• Project 01 – Due Monday 10/09
  • Implement Blackjack!
  • Paired assignment – can work with a partner

• Midterm 1 – Thursday 02/12
Announcements - Collaboration Policy

Discuss approaches to problems, and to sketch out general solutions

MUST write up the homework answers, solutions, and programs individually without sharing specific details, i.e. code
Announcements – Collaboration Policy

Gradescope automatically compares your assignments and gives a similarity score (it’s a percentage).

If the percentage is very high, it’s a sign that code was shared

First offense:
  0 points on that portion of the assignment
Second offense:
  0 points on the assignment
Third offense:
  Let’s not get there
Agenda

Recursion - review
Arrays – reviews
Misc for Blackjack
Strings and Arrays as Objects
Recursion Example – printVowels

Write a recursive function that prints just the vowels in a String
User Input

Ask a user for their first and last name

If the user doesn’t give a first and last name, prompt them again
Recursion limitations

• Limited number of times we can recurse
  • Stackoverflow – too many frames

• Potentially memory inefficient
  • If we copy data in subproblems – we’ll worry about this in a few weeks

• Performance: might duplicate unnecessary work
  • We’ll define performance later in the semester
Arrays
Arrays

Idea: Store multiple values into a single variable

Values are sequential

Analogous to a list
Arrays

double val = 3.0;

double[] vals = {3.0, 6.0, 7.0, -2.5};
Arrays

Three ways to initialize an array

1. With an initial value
   ```java
   int[] numbers = {1, 2, 5};
   ```

2. With allocated space, but uninitialized
   ```java
   int[] numbers = new int[3];
   ```

3. With an empty array reference
   ```java
   int[] numbers = null;
   ```
Array Indexing

Access individual elements of an array with indexing

array[index]

We use zero-based indexing

first element is 0
last element is length-1

Accessing indices out of range results in a runtime error!
Recursion Example – printList

Write a recursive function that prints the contents of an array
Command line arguments

```java
public static void main(String[] args)
```

Command line arguments are an *array of String*

Exercise: Write a program called `commandLineArgs.java` that

1) prints out 3 command line arguments that are passed in.
2) Compute the sum of three command line arguments (assuming they are integers)
Random from Range

Say we want to choose a random number between 100-200 (exclusive)

Math.random() creates a random double between 0 and 1

Multiply that by the range (max – min)

Add the result to the min value
Agenda

Recursion - review
Arrays – reviews
Misc for Blackjack

Strings and Arrays as Objects
Initializing empty arrays

```java
int[] nums = new int[3];
    [0, 0, 0]

String[] strs = new String[3];
    [null, null, null]
```
What is nums after we call add1?

```java
public static void add1(int[] list, int pos) {
    if (pos >= list.length) {
        return;
    }
    list[pos] += 1;
    add1(list, pos+1);
}

public static void add1(int[] list) {
    add1(list, 0);
}

public static void main(String[] args) {
    int[] nums = {10, 20, 30};
    printList(nums);
    add1(nums);
    printList(nums);
}```
Objects

Strings and arrays are **NOT** primitives

They are objects

Explains why we can’t use “==” to compare Strings

“==” checks if two objects are the same
not if the two values are the same
2-D Arrays – Arrays of Arrays