– CS 111 –

Week 6: [Input] and [Output]

Questions?

If you need help, there's always the CSL and RU CATS!

(Hill Center 252) (Hill Center 248)

Are you looking for a chance to…

- apply what you've learned in CS111 so far
- learn how to make fun projects w/ the help of mentors
- build your resume, and get it reviewed by upperclassmen

- win prizes from a $5000+ prize pool
- take home **free merch**, enjoy **free food**, and **make new friends**?

Register for ***HackRU!!***

100% beginner friendly hackathon, on campus!

hackru.org

HACKRU

FALL 24 MEOW

hackru.org



A 24-hour hackathon where ANY student can:

- learn how to code in a beginner friendly environment
- build an awesome project with mentors & peers
- boost their resume with experience
- win prizes from our $5000+ prize pool

+ free food, free shirts, & free swag!

When: October 26-27, 2024
Where: College Avenue Student Center
Scan the QR Code to find out more! →

RUTGERS–NEW BRUNSWICK
School of Arts and Sciences

Kahoot!

Due Dates:

- Wednesday, 10/23: RU Bus Stop and Egyptian Pyramid

Study Groups!

* Monday 3:50-5:10, Serc 106
* Tuesday 2:00 - 3:20, AB 1150B (CA Learning Center)
*Tuesday 5:40 - 7:00, AB1150A (CA Learning Center)
* Wednesday 10:20-11:40, Serc106 (Busch learning center)

Set Up

- http://tinyurl.com/111inputoutput
  - Contains a link for downloading the StdLib Library
    - Library = collection of classes for you to use!
  - Contains a link to a JavaDoc
    - Java Documentation → Your user manual!
  - Contains some files you'll find handy for this week's problems

- You NEED to have the StdLib library setup for class

Problem Set

Warm up

Discuss:

1. How do we “**import**” libraries so that they can be used in our program?

2. Once we have a library, how can we use it?

3. What are some benefits of libraries?

Warm Up Soln.

Solution:

1. Drag the lib files directly to the folder

2. Classname.methodName(<any parameters>)
   Ex: StdDraw.square(5, 5, 7);

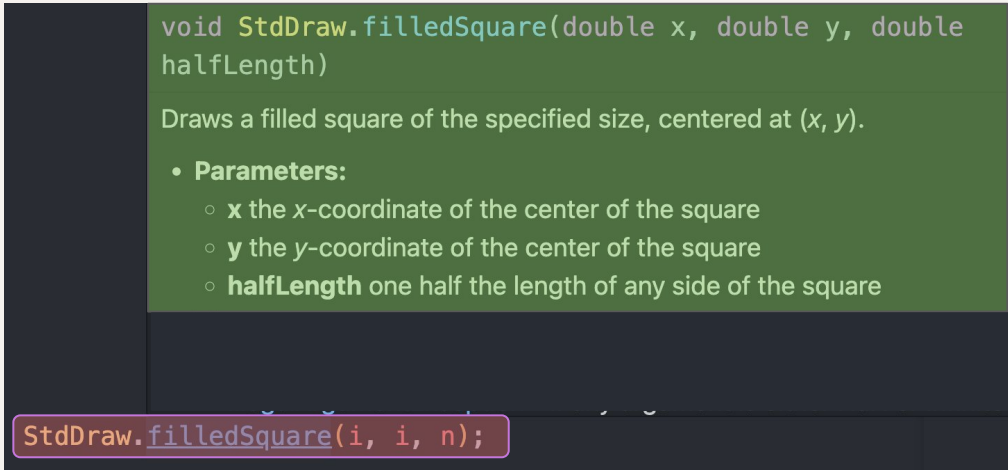
3. They make our coding easier! For example, with StdDraw, we simply call methods and they magically appear on the screen :>

Using a Method

1. Most methods usually have some kind of *documentation* – after you type the method name in VSCode, hover over it to see a short description. (figure 1a)

2. To use a method, follow the following syntax
   Classname.methodname(param1, param2)

3. Ex: StdDraw.filledSquare(0.5, 0.5, 0.2);
   a. Draws a filled in square at (0.5, 0.5), with side length of 0.4



Easier Problems

Guessing Game!

Create a Program that implements a guessing game via StdIn.

Generate a random int between 1 to 100, and prompt the user to try to guess what the number is.

- As the user enters numbers, the program should tell whether the guess is too high or too low before allowing them to input another guess

Hint: Use readInt() from the StdIn library!

Guessing Game Soln.

```
int secretNumber = 1 + (int)(Math.random()*100);
StdOut.println("I'm thinking of a number between 1 and 100");
int guess = 0;
while(guess!=secretNumber){
    StdOut.print("What's your guess? ");
    guess = StdIn.readInt();
    if(guess == secretNumber){
        StdOut.println("You win!");
    }
    if(guess < secretNumber){
        StdOut.println("Too low");
    }
    if(guess > secretNumber){
        StdOut.println("Too high");
    }
}
```

Calculator!

Create a Program that functions as a basic arithmetic calculator, the program should ask the user to input two integers, and ask for an arithmetic operator (“+”, ”-”, ”*”, ”/”). It will then perform the corresponding arithmetic operation and display the result

- Keep in mind, error conditions
- What happens if you divide something by 0

All input and output must be done using StdIn and StdOut libraries

Hint: To compare what's inside strings, use (StringName).equals!

```
> str1.equals(str2)
```

Calculator Soln.

```
public class Calculator{
    public static void main(String[] args) {
        StdOut.print("Enter the first integer: ");
        int num1 = StdIn.readInt();

        StdOut.print("\nEnter the second integer: ");
        int num2 = StdIn.readInt();

        StdOut.print("\nEnter an arithmetic operator (+, -, *, /): ");

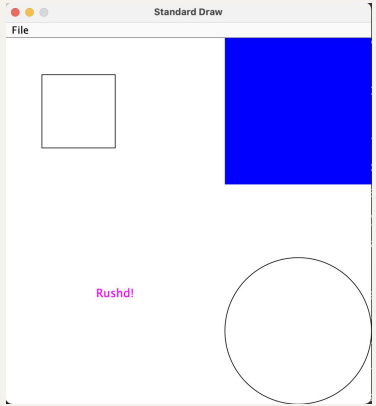
        String operator = StdIn.readString();

        if (operator.equals("+"))
            StdOut.println("Result: " + (num1 + num2));
        else if (operator.equals("-"))
            StdOut.println("Result: " + (num1 - num2));
        else if (operator.equals("*"))
            StdOut.println("Result: " + (num1 * num2));
        else if (operator.equals("/"))
            if (num2 == 0)
                StdOut.println("Error: Division by zero is not allowed.");
            else
                StdOut.println("Result: " + ((double) num1 / num2));
        else
            StdOut.println("Error: Invalid operator.");
    }
}
```

Drawing Shapes

Write a program that **draws any 2** of these, we've given where to center the objects for you :p

- non-filled square
  - x = 0.2, y = 0.8
- non-filled circle
  - x=0.8, y=0.2
- BLUE filled square:
  - x=0.8, y=0.8
- Any color text:
  - x=0.3, y=0.3



StdDraw Methods Reference:

- **public static void square(double x, double y, double halfLength)**
  - Draws a square of the specified size, centered at (x, y).
  - Parameters:
    - x - the x-coordinate of the center of the square
    - y - the y-coordinate of the center of the square
    - halfLength - one half the length of any side of the square
- **public static void filledSquare(double x, double y, double halfLength)**
  - Draws a filled square of the specified size, centered at (x, y).
  - Parameters:
    - x - the x-coordinate of the center of the square
    - y - the y-coordinate of the center of the square
    - halfLength - one half the length of any side of the square
- **public static void circle(double x, double y, double radius)**
  - Draws a circle of the specified radius, centered at (x, y).
  - Parameters:
    - x - the x-coordinate of the center of the circle
    - y - the y-coordinate of the center of the circle
    - radius - the radius of the circle
- **public static void text(double x, double y, String text)**
  - Writes the given text string in the current font, centered at (x, y).
  - Parameters:
    - x - the center x-coordinate of the text
    - y - the center y-coordinate of the text
    - text - the text to write
- **public static void setPenColor(StdDraw.COLOR)**
  - Sets the pen color for whatever will be drawn.
  - Instead of COLOR do any color such as BLUE or RED

Drawing Shapes Sol.

```
public static void main(String[] args){
        StdDraw.square(.2,.8,.1);
        StdDraw.circle(.8,.2,.2);
        StdDraw.setPenColor(StdDraw.BLUE);
        StdDraw.filledSquare(.8,.8,.2);
        StdDraw.setPenColor(StdDraw.MAGENTA);
        StdDraw.text(0.3, 0.3, "Rushd!");
  }
```

Less… easier Problems

Nested Circles

Write a program that reads an integer number n from the command line and draws n circles with alternative colored circles nested in each other. For example, for n equals to 5:

- **public static void filledCircle(double x, double y, double radius)**
  - Draws a filled-in circle of the specified radius, centered at (x, y).
  - Parameters:
    - x - the x-coordinate of the center of the circle
    - y - the y-coordinate of the center of the circle
    - radius - the radius of the circle

Example:
$ javac NestedCircle.java
$ java NestedCircle 5



Nested Circles Soln.

```
public class NestedCircle {
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        double r = 0.5;
        double diff = r/n;
        for (int i = 1; i <= n; i++) {
            if(i%2 == 0)
                StdDraw.setPenColor(StdDraw.BLACK);
            else
                StdDraw.setPenColor(StdDraw.GRAY);
            StdDraw.filledCircle(0.5, 0.5, r);
            r -= diff;
        }
    }
}
```

Students



Write a program that reads a file, studentinfo.txt, and stores the data (hint: arrays). You should be able to

1) Print a list of email address of students in sections 4 and 5.
2) Display all students as Last, First
3) Challenge: Use only **one** array

Your execution should look something like this:

```
java Students < students.txt
```

Feel free to display output as you see fit.

Students Soln.

```
public class students {
   public static void main(String[] args) {
       int numStudents = StdIn.readInt();
       String [] studentFirst = new String[numStudents];
       String [] studentLast = new String[numStudents];
       String [] studentID = new String[numStudents];
       int [] studentSec = new int[numStudents];
       for(int i = 0; i < numStudents; i++){
           studentFirst[i] = StdIn.readString();//first
           studentLast[i] = StdIn.readString();//last
           studentID[i] = StdIn.readString();//netID
           studentSec[i] = StdIn.readInt();//section
       }

       for(int i = 0; i < numStudents; i++)
           if(studentSec[i] == 4 || studentSec[i] == 5)
               StdOut.println(studentID[i]);

       for(int i = 0; i < numStudents; i++)
           StdOut.println(studentLast[i] + ", " + studentFirst[i]);
   }
}
```



Next Slides:

Challenge Intuition,

Challenge Soln



IDEA: i-based indexing (student i + some constant)

i	i+1	i+2	i+3	i	i+1	i+2	i+3	i	i+1	i+2	i+3
0	1	2	3	4	5	6	7	8	9	10	11
FIRST	LAST	NETID	EMAIL	FIRST	LAST	EMAIL	NETID	FIRST	LAST	EMAIL	NETID
0	0	0	0	1	1	1	1	2	2	2	2



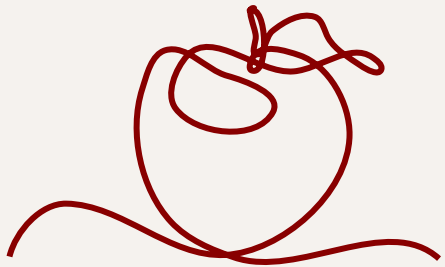
Student 0 (0,1,2,3)

Student 1 (4,5,6,7)

Student 2 (8,9,10,11)

Challenge Soln.

```
    int numStudents = StdIn.readInt();
    String [] students = new String[numStudents * 4];
    for(int i = 0; i < numStudents; i++){
        students[i*4] = StdIn.readString();//first
        students[i*4 + 1] = StdIn.readString();//last
        students[i*4 + 2] = StdIn.readString();//netID
        students[i*4 + 3] = StdIn.readString();//section
    }
    for(int i = 0; i < numStudents * 4; i += 4){
        if(students[i+3].equals("4") ||
            students[i+3].equals("5")
        ){
            StdOut.println(students[i+2]);
        }
    }
    for(int i = 0; i < numStudents; i++){
        StdOut.println(students[i*4 + 1] + ", " + students[i*4 +0]);
  }
}
```





Mad Libs!

Mad Libs!
The goal is to fill in the following story using user input:

_____ (name 1) went on a walk! On the walk, they found a _____ (noun1) that was super valuable because it was so _____ (adj1) ! They went to their neighbor, _____ (name 2) and tried to sell it to them for a crazy amount of _____ (number 1). They closed the deal, and _____ (name 1) used it to buy a super ____ (adj2) ____ (noun 2).

Feel free to copy paste the template from tinyurl.com/111inputout:

1) Use StdIn to prompt the user for their “madlibs”
2) Fill in the story and display it to the user via StdOut
3) Save the outputs to a new textfile, rather than having it displayed in terminal
4) Download the madlibinput.txt file, and use piping in order to pass the inputs into the program.
5) Feel free to customize the madlib once you’re done!

Mad Libs Solution

```
//headers omitted
        System.out.print("Enter a name: ");
        String name1 = StdIn.readString();
        System.out.print("Enter another name: ");
        String name2 = StdIn.readString();
        System.out.print("Enter an adjective: ");
        // rest of the inputs omitted for brevity's sake, same pattern tho
        System.out.print("Enter a number: ");
        int num1 = StdIn.readInt();
        // Fill in the mad libs story
        String madLibsStory = name1 + " went on a walk! On the walk, they found a
           " + noun1 + " that was super valuable because it was so " + adj1 + "!
           They went to their neighbor, " + name2 + ", and tried to sell it to
           them for a crazy amount of " + num1 + ". They closed the deal, and " +
           name1 + " used it to buy a super " + adj2 + " " + noun2 + ".";

        StdOut.print(madLibsStory);
```

