Secctio-3

of 1

                     Automatic Zoom                     Actual Size                     Page Fit                     Page Width                                          50%                     75%                     100%                     125%                     150%                     200%                     300%                     400%

Copyright © 2022, Oracle and/or its affiliates. Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.  
Java Foundations  
Practices - Section 3  
Creating a JavaLibs Game  
Overview  
Section 3 has shown you everything you’ll need to recreate a  
JavaLibs program, similar to the one you played at the beginning  
of the section. It may take a bit of thinking, but it’s a challenge  
you’re definitely ready for. Your program requirements are  
described below. Good luck, and have fun!  
Tasks  
Your goal is to create a program similar to JavaLibs. Write a story where certain parts of the resulting story text are modified by the  
user’s input. Prompt the user for various inputs.  
You may accept user any number of ways, including a JOptionPane, or Scanner input from the console. However, choose only one  
method. Don’t use multiple methods of accepting input. Similarly, if you use JOptionPane to get input, use JOptionPane to show  
the resulting story.  
When you output your story, make sure your all your text is visible at the same time. It’s not ok for text to be too long for your computer  
screen or output window. Your story will need to be spread across several lines instead of being printed in one giant line of output.  
This helps keep your output clean and your program more user friendly.  
It’s ok for your program to crash if the user inputs inappropriate data. In other words, It’s ok if your program crashes because you’ve  
expected the user to input a number, when they’ve instead input a String. We’ll cover exception handling later in the course.  
Your program must also do the following:  
• Accept at least 1 input, to be parsed as a String  
• Accept at least 1 input, to be parsed as an int  
• Accept at least 1 input, to be parsed as a double  
• Use at least 1 input in a question for the user  
• Do math with at least 1 int input  
• Do math with at least 1 double input  
• Accept at least 10 total inputs  
It’s ok for this problem set to write your entire program within the main method.  
The JavaLibsPractice.java file is available to help you get started.

Code: package snakebox;

// Snake.java

import java.util.Scanner;

public class JavaLibsPractice {

public static void main(String[] args) {

// Accepting at least 10 inputs

try (Scanner scanner = new Scanner(System.in)) {

// Accepting at least 10 inputs

System.out.println("Welcome to the JavaLibs game! Please enter the following details:");

// Accepting String inputs

System.out.print("Enter your name: ");

String name = scanner.nextLine();

System.out.print("Enter your favorite color: ");

String color = scanner.nextLine();

System.out.print("Enter your hometown: ");

String hometown = scanner.nextLine();

System.out.print("Enter your favorite food: ");

String food = scanner.nextLine();

// Accepting int input

System.out.print("Enter your age: ");

int age = scanner.nextInt();

// Accepting double input

System.out.print("Enter your height in meters: ");

double height = scanner.nextDouble();

System.out.print("Enter a random number: ");

int randomNumber = scanner.nextInt();

System.out.print("Enter the price of your favorite snack: ");

double snackPrice = scanner.nextDouble();

// Additional inputs to meet the 10 input requirement

System.out.print("Enter your favorite animal: ");

String animal = scanner.next();

System.out.print("Enter your dream destination: ");

String destination = scanner.next();

// Performing calculations

int futureAge = age + 5;

double doubleHeight = height \* 2;

// Creating the story

String story = String.format("Hello, %s! Here is your personalized story:\n", name);

story += String.format("Once upon a time in %s, there lived a person named %s who loved the color %s.\n", hometown, name, color);

story += String.format("%s was %d years old and %f meters tall.\n", name, age, height);

story += String.format("Every day, %s would eat %s for lunch and dream about visiting %s.\n", name, food, destination);

story += String.format("One day, %s found a magical %s that granted %d wishes.\n", name, animal, randomNumber);

story += String.format("The first wish was for an endless supply of %s, which costs %f each.\n", food, snackPrice);

story += String.format("In five years, %s will be %d years old and %f meters tall.\n", name, futureAge, doubleHeight);

story += String.format("And they all lived happily ever after in %s.\n", destination);

// Displaying the story

System.out.println("\n" + story);

// Closing the scanner

       }

    }

}

Output:

