

## Lab I: Building a Game Box

### This lab holds six goals:

1. Help you practice the dynamics of a technical interview.
2. Apply your Java and problem-solving skills.
3. Deal with abstraction to design your own solutions to a problem.
4. Reflect on different ways of solving the same problem.
5. Have fun.
6. **Launch your Dev Journal.**

### 1. Behavioral Instructions

Acknowledgement: Dr. Elliott designed this part inspired by a chat with Dr. Sarah Barks.

- a. You and your peer will take turns coding a solution to the Game Box.
- b. The person in charge of coding a solution will do it as they explain why and what is being coded.
- c. The person in charge of listening will act as an interviewer and ask clarifying questions.
- d. Before each turn, exchange helpful and kind feedback. The goal is to *encourage* each other to feel confident and prepared for a technical interview.
- e. Remember to compile and document your thoughts into your Dev Journal.

### 2. Coding Policy

1. A note on using Grammar Checker tools such as Grammarly to answer textual questions: It is fine to use those as long as you DO NOT make use of generative AI features.
2. You will read and follow our [Academic Integrity Policy](#).
3. You will follow our course's [course coding policy](#).
4. **Nope, you are not allowed to use AI -Powered Code Completion Tools** in this assignment.
5. Read: [Can You Explain Your Code?](#)

### 2. Problem Description - You will build a "Game Box"

The Game Box allows users to choose and play a game from a set of 15 options. The games are:

- 1) Rock, paper, scissors. Your program must be biased to picking "rock" a bit more than the other options.
  - a. You will come up with your own version of designing a biased program.
  - b. Reflect: How good is your design? What other designs are possible? For someone testing your code, how easily would they notice this bias? What ethical considerations should there be in making it easier or harder to identify the bias?)
- 2) Guess a number using a binary search strategy. The program asks someone to pick a whole number between 1 and 100. Then, *the program* tries to guess that number.
- 3) You are **NOT** allowed to use either Arrays or Strings!!! Write a program that verifies if a given number  $n$  is a Palindromic number. Palindromic numbers are "a number (such as 16461) that remains the same when its digits are reversed" – [Wikipedia](#).
- 4) Write a program that verifies if a given number  $n$  is a Palindromic number. This time, using arrays.

- 5) Given a number  $n$ , your program must print the sum of the first  $n$  positive numbers. In addition, your program must print the first " $n$ " positive and even numbers. (Example: if  $n=3$ , the program will print: "The sum of the 3 first positive numbers is 6"; and then: "The first 3 positive and even numbers are 2, 4, 6").
- 6) Given a number  $n$ , your program must calculate and print  $n!$  (What kinds of numbers are valid for this problem?)
- 7) Your program will calculate and print the first  $n$  Fibonacci numbers.
- 8) You will build a program that asks for a temperature in Fahrenheit, converts it to Celsius, and prints it.
- 9) You will build a program that asks a user to provide a distance in kilometers and the total elapsed time in hours. Your program will print the average speed in kilometers per hour.
- 10) NOTE: You are NOT allowed to use  $\%$  (modulo operator) to get the remainder. Your program will get a 3-digit number from user input. Then, it will calculate the number's remainder from a division by 10. Your program will print: the remainder and the other two digits from the input number. Example: if a user types 193, your program will print: "the remainder is 3, and the other two units in 193 are 19".
- 11) Given  $n$  and two positive integers  $i$  and  $j$ , your program will print, in ascending order, the  $n$  first positive numbers that are multiples of  $i$ , or  $j$ , or both. (You are not allowed to use the modulo operator  $\%$ ). Example: for  $n=6$ ,  $i=2$ , and  $j=3$ , your program should print: 0, 2, 3, 4, 6, 8.
- 12) Your program will ask a number from the user and then verify if the number is perfect. Example: 6 is a perfect number since  $1 + 2 + 3 = 6$ . Run your program to find out: Is 496 a perfect number? (You are not allowed to use the modulo operator  $\%$ )
- 13) Two integers  $a$  and  $b$ , are congruent modulo  $n$  if they have the same remainder when divided by  $n$ . Given the positive integers  $a$ ,  $b$ , and  $n$ , your program will print if  $a$  and  $b$  are congruent modulo  $n$ . Hint: you can use  $a = 35$ ;  $b = 39$ ;  $n = 4$ . (You are not allowed to use the modulo operator  $\%$ ).
- 14) Your program will ask the user to provide an integer  $n$ . Given  $n$ , your program will check if it is the result of three consecutive integers multiplied. Example: 504 should print "yes, the provided number is the result of three consecutive integers multiplied, since  $7 * 8 * 9 = 504$ ".
- 15) You will come up with your own game to be added as the last game in the Game Box.

### 3. Additional Instructions

Remember to compile and document your thoughts into your Dev Journal.

Your program must start and prompt the user to choose whether to run the driver or interact with the program.

The driver allows someone to check your program's functioning without interacting with it. That means you should implement a driver with predefined variables so that the program runs three different tests for each game and produces output informative enough for someone to check its functioning. In that sense, your program must accomplish two steps:

1. Print what is happening on the screen and

2. Write that same info in a text file.

#### 4. What to hand?

You will upload a pdf file only. In the pdf, the *team* will answer the questions below. Teams must collaboratively work on every single problem and questions – splitting tasks is not allowed.

- **You must use either Verdana, Tahoma, or Palatino Linotype (font-size 12 or 14).**

1. (1.0) How was the experience of following the behavioral instructions?
2. (1.0) Explain how your program fulfills the requirement of being biased to picking “rock” more than the other options (in the rock paper scissors option).
  - a. This question asked you a few reflection questions. After reflecting, answer: how good is your design?
3. (1.0) What process did you follow to decide what to add to your Dev Journal? How was the process of launching your dev journal? **Important:** Each team member must identify themselves and answer this question. If your team has three people, I expect to see three answers properly identified.
4. (2.0) You will thoroughly explain how your driver works (including chosen parameters and why).
5. (1.0) You will add screenshots of your driver’s output to contextualize the previous question.
  - a. Make sure to add pictures that are easy to see!
  - b. Add explanatory, comprehensive captions below your images.
6. (1.0) How would you have coded a solution to the “Guess a number” if not instructed to use a binary search strategy? Explain (even if you would have used that strategy anyway).
7. (1.0) What game did you add as the last option within your Game Box? Why did you choose this game?
8. (1.0) How was the experience of solving the palindromic question using two different approaches?
9. (1.0) You will ask at least two people to interact with your Game Box and provide you with feedback (note that all games must be tested, not necessarily by the same person). Now, explain what feedback you got and reflect upon it.

#### 5. Policy.

- a. For team submissions, only a single submission per team. (Make sure to add all team members!!!!)

**Grading Scheme – Each question will be graded according to the rubric below:**

- For team submissions, only a single submission per team. (Make sure to add all team members!!!!)

100%	70%	60%	40%	20%
1. All parts of the question are correctly answered. 2. Answer is comprehensive and shows a great level of reflection.	1. All parts of the question are correctly answered. 2. Answer is not comprehensive.	1. Most parts of the question are correctly answered.	1. Some parts of the question are correctly answered.	1. Very few parts of the question are correctly answered.

**Acknowledgements:** a few of these questions were inspired by the “Caderno de exercicios, IME USP, thirteenth edition, 2005. (Saudoso professor **Ernesto G. Birgin!**)