**The Great Computer Challenge, 2021**

***Object Oriented Programming, Level 4***

# **Challenge 1**

# **Background**

A palindrome is a word, phrase, or sentence that is spelled the same way forward and backward.  Some examples of palindromes are “radar,” “able was i ere i saw elba” and (if spaces are ignored) “a man a plan a canal panama.”

# **Guidelines & Requirements**

Write a program that reads an input string, checks if it’s a palindrome, and outputs TRUE if it’s a palindrome or FALSE if otherwise.   Your program should ignore case, spaces, and punctuation in the input string.

# **Palindromes**

A palindrome can also be number, phrase, or other sequence of characters which reads the same backward as forward, such as madam or racecar. There are also numeric palindromes, including date/time stamps using short digits 11/11/11 11:11 and long digits 02/02/2020. Sentence-length palindromes ignore capitalization, punctuation, and word boundaries.

# **Judging Criteria**

Your program should ignore case, spaces, and punctuation in the input string.

# **SOL Correlation**

1. The student will create programs demonstrating an understanding of the interactions between classes in object-oriented design, and by implementing classes with instance data and methods to satisfy a design specification.

# **Challenge 2**

# **Background**

The COVID-19 pandemic has had a significant impact on the housing market in Hampton Roads.  There are fewer houses on the market because fewer sellers are willing to list their properties or allow strangers to enter their homes because of health concerns.  In spite of the coronavirus turmoil, home buying activity is still strong.

Your real estate investment company is looking to aggressively purchase homes in this area to flip or lease.  As a first step in developing a custom application to list, buy, sell, and lease real estate, you plan on creating a GUI app to compare monthly mortgage payments for standard and custom loans.

# **Guidelines & Requirements**

Your app must allow the user to enter a loan amount, loan duration or term (in years), and an *annual* interest rate. Based on these values, your app should calculate and display 1) loan amount, 2) annual interest rate, 3) loan term in years 4) monthly payment, and 5) total interest paid for the custom loan duration as well as 10-, 15- and 30-year loans. You may use a slider to select the custom loan duration.

# **Mortgage Calculator**

The **formula** to calculate the monthly payment is

M= P[r(1+r)^n/((1+r)^n)-1)]

M = the total monthly mortgage payment.

P = the principal loan amount.

r = *monthly* interest rate. To get the monthly rate, divide the annual interest rate by 12 (the number of months in a year)

n = number of payments over the loan’s lifetime. Multiply the number of years in your loan term by 12 (the number of months in a year) to get the number of payments for your loan. For example, a 30-year fixed mortgage would have 360 payments (30x12=360).

**Sample calculations for given input**

**Loan amount:** $150,000

**Annual interest rate:** 4.5%

**Loan Term:** 7 years

|  |  |  |
| --- | --- | --- |
| **Loan Term (years)**  | **Monthly Payments ($)**  | **Total Interest Paid ($)**  |
| 7  | 2.085.02  | 25.142.03  |
| 10  | 1,554.58  | 36,549.14  |
| 15  | 1,147.49  | 56,548.19  |
| 30  | 760.03  | 123,610.07  |

(25 points)

# **Judging Criteria**

Your app must allow the user to enter a loan amount, loan duration or term (in years), and an *annual* interest rate. Based on these values, your app should calculate and display 1) loan amount, 2) annual interest rate, 3) loan term in years 4) monthly payment, and 5) total interest paid for the custom loan duration as well as 10-, 15- and 30-year loans. You may use a slider to select the custom loan duration.

# **SOL Correlation**

1. The student will implement programs that accept input from a variety of sources and produce output based on that input.

# **Challenge 3**

# **Background**

Neighbor, based in Salt Lake City, connects homeowners online with people who need storage space. The premise is pretty simple, it’s pretty much like Airbnb, just with stuff instead of people. Homeowners can advertise space and set their own prices, and renters pay money to store their items.

# **Guidelines & Requirements**

Required information for a homeowner to sign up is first and last name and a valid telephone number. Once they sign up, homeowners should be able to create multiple listings of their storage spaces by entering the address and the cubic footage available of each storage space.

Renters sign up by entering their first and last name and a valid telephone number. Once they sign up, renters should see a list of available storage spaces and be allowed to “request” that space by entering the items they wish to store there. Once requested, a space should not appear on the available storage spaces.

Homeowners can accept or reject an offer to rent a storage space. When they accept an offer, the homeowner should be shown the name and telephone owner of the renter. If an offer is rejected, the space should reappear on the list of available storage spaces.

(50 points)

# **Neighbor Self-Storage**

Your task is to write a GUI program that allows (1) a homeowner to sign up and list their storage space for rent, (2) a renter to sign up and request a storage from the homeowners, and (3) a homeowner to accept a rental contract.

# **Judging Criteria**

Your task is to write a GUI program that allows (1) a homeowner to sign up and list their storage space for rent, (2) a renter to sign up and request a storage from the homeowners, and (3) a homeowner to accept a rental contract.

# **SOL Correlation**

1. The student will implement programs that accept input from a variety of sources and produce output based on that input.

*Have fun and thanks for participating in the Great Computer Challenge, 2021!*