-1) Team members. Provide the names and majors of the team members.
Julio Rivera, Computer Engineering Major
Jacob Braden, Computer Science Major
Caleb Cope, Cybersecurity Engineering Major
Casen Woody, Computer Science Major.

2) Description.
This project was performed to complete the original Monopoly game in a digital format. This included a main menu, player menu to enter player names, and the main board game. Players can play the rules similarly to the original game. There are some differences in the rules.

3) Analysis
a. Sequence of interactions include:
User clicks “Start” button
[-> : Navigates to another screen]
—----------
[Enter player names] [User input] [User clicks “Start”]->
[Now in game][Users roll dice][User with highest roll goes first, Lowest goes last]->
[User lands on spaces] [User buys property/gets a card] [Action is performed by the program]->
[Game continues until 3 of 4 players goes bankrupt]
—----------
[Quit]->
[Application exits]
—----------

b. Essential and enhancement items.

Essential:	Enhancement:
Random Movement	Trading
Properties	Auctioning
Going Bankrupt	Music/Audio
Chance Cards	
Currency	
Jail	
Community Chest	
Bank	

We started with 134 work hours and ended with about 20 hours remaining.

4) Design.
The software is composed of:
Classes:
Bank
Property extended by: ColoredProperty, UtilityProperty
Space extended by: JailSpace, GoSpace, ChanceSpace, CommunityChestSpace, FreeParkingSpace, GoToJailSpace, IncomeTaxSpace, JustVisitingSpace, LuxuryTaxSpace, and PropertySpace.
Player

These elements will allow the software to keep track of player information such as owned properties, username, currency, and location on the board. These will allow the program to keep track of what properties are owned, who is bankrupt, and what cards to give the player. These will allow the software to send players to jail and other places around the board when required.

b. The domain model. A presentation of the essential items in the domain including the logic of the items and their interaction. In addition to descriptive paragraphs, provide a listing and description of the items based on noun/verb separation, CRC cards, or UML diagrams. The domain model should not contain references to the GUI in terms of what is displayed to the user or what the user inputs. Think of this as a collection of the things in the domain and what they do. Your design should indicate the public methods for the classes. You can base this on your previous reports and should reference the CRCcards or UML diagrams. The actual cards and diagrams may be placed in appendices.

—----------

1. Class Player:
    a. Responsibilities:
        i. Keep Track of each player's money, position, name, token, ID, and inventory
        ii. Roll dice
        iii. Handle movement
        iv. Purchase properties & buildings
        v. Pay rent
        vi. Sell buildings
        vii. Call bankruptcy
        viii. Receive & pay money to the bank
        ix. Mortgage properties
    b. Collaborators:
        i. Board

        ii. Space
        iii. Bank
        iv. Property
2. Class boardController:
    a. Responsibilities:
        i. Keep track of visuals & UI components of tokens and buildings on the board
        ii. Update visuals
    b. Collaborators:
        i. Player
        ii. Space
3. Class Space:
    a. Responsibilities:
        i. Keep track of each space's name and position
        ii. Handle the event when a player lands on it
    b. Collaborators:
        i. Board
        ii. Player
        iii. GoSpace
        iv. JustVisitingSpace
        v. JailSpace
        vi. FreeParkingSpace
        vii. GoToJailSpace
        viii. ChanceSpace
        ix. CommunityChestSpace
        x. IncomeTaxSpace
        xi. LuxuryTaxSpace
        xii. PropertySpace
        xiii. Bank
4. Class Bank:
    a. Responsibilities:
        i. Keep track of unowned properties, buildings, and cards
        ii. Create all properties, buildings, and decks
    b. Collaborators:
        i. Player
        ii. Property
        iii. Card

5. Class Property:
    a. Responsibilities:
        i. Keep track of a property's cost, position, name, ownership, and mortgage status
        ii. Change ownership
        iii. Calculate rent
        iv. Mortgage
    b. Collaborators:
        i. Player
        ii. PropertySpace
        iii. Bank
        iv. ColoredProperty
        v. RailroadProperty
        vi. UtilityProperty
6. Class GoSpace:
    a. Responsibilities:
        i. Set space name & position to "Go" & 0
    b. Collaborators:
        i. Space
        ii. Player
7. Class JustVisitingSpace:
    a. Responsibilities:
        i. Set space name & position to "Just Visiting" & 10
    b. Collaborators:
        i. Space
        ii. Player
8. Class JailSpace:
    a. Responsibilities:
        i. Set space name & position to "Jail" & 40
    b. Collaborators:
        i. Space
        ii. Player
9. Class FreeParkingSpace:
    a. Responsibilities:
        i. Set space name & position to "Free Parking" & 20
    b. Collaborators:
        i. Space

        ii. Player
10. Class GoToJailSpace:
    a. Responsibilities:
        i. Set space name & position to “Go to Jail” & 30
        ii. Put player in jail
        iii. End turn immediately
    b. Collaborators:
        i. Space
        ii. Player
11. Class ChanceSpace:
    a. Responsibilities:
        i. Set space name & position to “Chance” & specified position
        ii. Draw & execute card from Chance deck
    b. Collaborators:
        i. Space
        ii. Player
12. Class CommunityChestSpace:
    a. Responsibilities:
        i. Set space name & position to “Community Chest” & specified position
        ii. Draw & execute card from Community Chest deck
    b. Collaborators:
        i. Space
        ii. Player
13. Class ChanceSpace:
    a. Responsibilities:
        i. Set space name & position to “Chance” & specified position
        ii. Draw & execute card from Chance deck
    b. Collaborators:
        i. Space
        ii. Player
14. Class IncomeTaxSpace:
    a. Responsibilities:
        i. Set space name & position to “Income Tax” & 4
        ii. Deduct $200 from the player’s total money
    b. Collaborators:
        i. Space

ii. Player

15. Class LuxuryTaxSpace:
    a. Responsibilities:
        i. Set space name & position to “Luxury Tax” & 38
        ii. Deduct $100 from the player’s total money
    b. Collaborators
        i. Space
        ii. Player
16. Class PropertySpace:
    a. Responsibilities:
        i. Set space name & position to match the given property
        ii. Depending on ownership: allow the player to the buy the property, start an auction, make the player pay rent to another player, or no event
    b. Collaborators:
        i. Space
        ii. Player
        iii. Property
17. Class ColoredProperty:
    a. Responsibilities:
        i. Keep track of building rents, color, amount of buildings, color set completion, and base rent
        ii. Calculate Rent
        iii. Add/Decrease buildings
    b. Collaborators:
        i. Property
        ii. Player
        iii. PropertySpace
        iv. Board
18. Class RailroadProperty:
    a. Responsibilities:
        i. Keep track of base rent
        ii. Calculate rent based on how many railroads are owned
    b. Collaborators:
        i. Property
        ii. Player
        iii. PropertySpace

19. Class UtilityProperty:
    a. Responsibilities:
        i. Calculate rent based on number of utilities owned & dice roll
    b. Collaborators:
        i. Property
        ii. Player
        iii. PropertySpace

—----------

c. Design elements for each of the required technical items. Keep these brief and simple. This is rather like the text that you would form from the CRC cards. Identify the technical item that you believe is your best work and explain why you think it is your best work. An initial frame might be “Element XXX has classes <set of classes> that do <what they do>. In particular class/package <particular class/package> is responsible for <responsibility>.” Feel free to edit. Note carefully any design patterns that you used.
i. Graphical user interface
ii. Text formatting and processing (html or other)
iii. Graphics sampled or constructive with manipulation or animation (could be all four).
iv. Storage and retrieval of information (flat file, database, XML or all of these).

5) Implementation
a. Overview. In this section provide a brief account of how the classes, interfaces, and enums are packaged. Present a brief sketch of what the API is for the project in terms of the classes, fields, and methods.. A starting frame might be “The implementation for <item> can be found in <elements>.” or “The implementation of the domain model is found in ClassA, ClassB, and ClassC in the model package.”
b. Public elements (API). List of the public classes, interfaces, and enums with the signatures for each public constructor, method, and field. For each of these, add a one line account of what it does and a one line account of how it is related to the design elements in section 4.

Player - playerSayHello, introduces player. rollTheDice(), rolls dice. playerPosition, handles movement. payMoney, receive and pay money to the bank.
Board - setPlayerName, sets player name. List_spaces, findCurrentPlayer, updateCurrentPlayer, updatePos, all keep track of and update player.
Space - handleEvent, keeps track of player position and information.
Chance Card and Community Space Card - handleEvent, will pick a card and handle information about it

Bank - createProperties, creates all properties in the bank.
Property - printInfo, creates and displays property information
IncomeTaxSpace - handleEvent pays 100 from player to bank
ColoredPropert, RailRoadProperty, UtilityProperty - calculateRent, calculates the rent of a space.
FreeParkingSpace, JailSpace, JustVisitingSpace - all handleEvent which manage basic info about the space.

6) Exercise and testing
a. Testing. A brief account about how you exercised or tested the code to determine whether it 'works'. This does not need to be the extensive testing that would occur in a "real" project. This is intended as an informal way of providing evidence that the code "works". You can note various exercise classes you used, the use of unit tests, or any other relevant materials.
We tested the software by running through the software from top to bottom as a user and making sure the functionalities work appropriately. We repeated this a few times amongst the team.
b. Walkthrough.
The User runs the program.
User clicks "Start" and is brought to the player name screen.
The user then enters the names of all four players, clicks "Start" and is brought to the board screen.
Once at this screen, users click on the roll dice button to get two randomly selected numbers that determine how far they move.
If the user lands on a property the click "Buy" and the property is added to their inventory.
The next user clicks the "Roll" button and lands on Chance or Community Chest space.
The program selects a random card to show the player and its effects are added to the player.
The program ends once 3 of the 4 players have gone bankrupt with the last player standing being the winner.

7) Evaluation
a. Use the materials in the text, especially chapter 3, and the class presentations as a guide for evaluation as appropriate.
Our software uses JavaFX to simulate a digital Monopoly game. Using most of our 130 work hours we managed to complete most of our goals.
b. Discuss how the essential and enhancement items were or were not accomplished.
We completed the essential items by dividing up the work at the beginning of the project where each team member had their own responsibilities. We seemed to use up more of our programming time as we got closer to the due date which slowed us down enough to where we did not include our enhancement items.
c. What was the best work?

The functionalities that were implemented with the board.
d. Discuss how the software could be improved:
The software could be improved through a more fun interactive experience. The board could have more visually appealing elements and visual pop-ups for different interactions. Such as a property pop-up when a property is purchased.
e. Discuss how the software could be extended.
The software could be extended by making the rules more complicated and adding animations based on the user's choices.

8) Submission notice
a. For this section have each person who worked on the team, sign with a submission date. This can be done in many ways. One way is to simply sign a piece of paper, take a picture with a phone, crop it, send it, and paste it into a document. For each such signature, type the person's name and date below the pasted in signature

Casen Woody 11/29/2022

Julio Rivera 11/29/2022

Jacob Braden 11/29/2022

Caleb Cope 11/29/2022