

**CS102 – Algorithms and Programming II**  
**Programming Assignment 1**  
**Spring 2025**

**ATTENTION:**

- Compress all of the Java program source files (.java) files into a single zip file.
- The name of the zip file should follow the below convention, where you replace the variables “Sec1”, “Surname” and “Name” with your actual section, surname and name:  
**CS102\_Sec1\_Asgn1\_Surname\_Name.zip**
- Upload the above zip file to Moodle by the deadline (if not significant points will be taken off). You will get a chance to update and improve your solution by consulting to the TAs and tutors during the lab. You have to make the preliminary submission before the lab day. After the TA checks your work in the lab, you will make your final submission. Even if your code does not change in between these two versions, you should make two submissions for each assignment.

**GRADING WARNING:**

- Please read the grading criteria provided on Moodle. The work must be done individually. Code sharing is strictly forbidden. We are using sophisticated tools to check the code similarities. The Honor Code specifies what you can and cannot do. Breaking the rules will result in disciplinary action.

**Simple Monopoly Game**

For this assignment, you will implement a simplified console-based Monopoly game, where the user can play against 1 to 3 computer players. At the beginning, the user determines the number of computer players, as well as their names. We should also enter the user's name. The game starts with each player having 10 coins. The playing order is determined randomly at the beginning of the game. The players roll a 6-sided die on their turn to determine how many spaces to move. All players start from the beginning location indicated with 0 in the map below. There can be multiple players in the same position.

0	A	B	C	1
L	Simple Monopoly Game			D
K				E
J				F
3	I	H	G	2

Each space in the game map is either a property that players can buy or corresponds to a special event. Properties are indicated with letters, and special events are indicated with numbers. The special events are as follows:

**0:** The players that pass through this position (by completing a tour on the map) are awarded with *three* coins; if they exactly end up on this cell, they are awarded with *six* coins instead.

**1:** The players that end up on this cell roll a 6-sided die to do one of the following:

1. Lose *two* coins.
2. Lose *one* coin.
3. Move *one* space forward.
4. Move *two* spaces forward.
5. Receive *one* coin and move *one* space forward.
6. Receive *two* coins and move *two* spaces forward.

If players move due to the special event, they can still buy the property they end up on (or need to pay the rent if it already belongs to someone else).

**2:** The players that end up on this cell receive *one* coin from each of the other players that still continue the game, so the other players each lose *one* coin, and this player receives those coins.

**3:** The players that end their movement on this cell will skip their next turn. They can roll and move again after skipping one turn.

Each property differs in price and rent. The property's rent is based on the number of houses on that property. At the beginning of the game, each property has zero houses. Each property can have at most *four* houses. Each property has a different cost for building one house. The rent increases as the number of houses increases. Any player that ends up on a cell owned by one of the other players should pay the rent to the owner. Each property has the following price, house cost and rent values:

Property	Price	Cost of building one house	Rent for different house counts
A, B, C	2	1	1 (0), 2 (1), 3, (2), 4 (3), 6 (4)
D, E, F	4	1	2 (0), 2 (1), 3, (2), 3 (3), 7 (4)
G, H, I	6	2	1 (0), 3 (1), 4, (2), 6 (3), 7 (4)
J, K, L	8	3	3 (0), 3 (1), 6, (2), 6 (3), 9 (4)

The house count for each rent amount is indicated in parentheses. For example, for property A, “1 (0)” means that rent is *one* coin when there are no houses, and “2 (1)” means the rent is *two* coins when there is *one* house on the property. The maximum rent for A is *six* coins, which happens when there are four houses on this property.

In each turn, the player should first roll the die and move accordingly; then, if the player’s move ends on a property that is not owned by any other player, it is possible to buy that property if the player has enough coins. If this player is the user, you should ask for the choice to buy or not; otherwise, the computer users should determine randomly. For computer players, you should indicate whether they buy the property with a message printed on the console. Also print the current coin amounts of the players after spending coins.

If the player’s movement ends on a special event cell, the corresponding event occurs based on the given descriptions. Finally, at the end of their turn, the players can build a house on one of their properties or sell one of their properties to the bank. If this player is the user, you should list the properties where building a new house is possible by checking the house cost and the player’s current coins. The user will choose one of the properties to build or may decide not to build on this turn. Computer players should randomly decide whether to build a house on one of their properties. Players should not attempt to build if they do not have enough coins. A player can’t build more than one house in one turn.

The players also have the option to sell one of their properties to the bank to receive its price back. In this case, any house on that property is demolished, and only the original property price is paid back by the bank. Then, that property belongs to no one and can be bought by any of the players if they end their movement on that cell. For example, suppose a player has property D with 3 houses on it; selling it to the bank demolishes the houses so that property D now has *zero* houses and pays *four* coins to the player so that this property now belongs to no player. Players cannot sell more than one property on their turn. Players either build a house or sell a property, they cannot do both in one turn. Computer players should choose a random property to sell if their coins drop below a certain threshold of your choice. Try to come up with a good strategy for the computer players to have a chance against the user.

When players end their movement on a property owned by some other player, they have to pay the owner the rent (calculated based on the number of houses on that property). If the player has insufficient coins to pay the rent, that player loses the game, and the rent is paid to the owner by the bank (the bank has unlimited coins). Even if the user loses the game, the computer players should continue until there is only one player left. There is also a 100-turn limit such that if nobody wins until the 100th turn, there is a tie between the continuing players, and the game ends. Players can also lose as a result of the special events if their coins drop below *zero*. Any player that lost the game should not roll or move and are removed from the map. Display the name of the winner if there is one, or print the remaining players at the end.

At the beginning of each turn, you should print the current map with each player indicated on the map with their initial letter. You should also indicate the number of houses for each property and

the owner if there are any. A sample map representation for the console is given below, but you are free to use different representations as long as you show the required information.

```
|0...|A.t0|B...|C.t1|1...|
|....|....|....|tn..|u...|
|L.n0|          |D.u1|
|....|          |....|
|K.n0|          |E...|
|....|          |....|
|J.n0|          |F...|
|....|          |b...|
|3...|I.t0|H...|G...|2...|
|....|....|....|....|....|
```

For example, this representation uses the following notation for each cell:

```
|X.yz|
|klmn|
```

X represents the cell name, this could be either a number for the special events {0, 1, 2, 3} or a letter for the property cells {A, B, C, D, E, F, G, H, I, J, K, L}. If this is a property, y shows the owner's letter or appears as a dot if this property is not owned by any player. For properties, z shows the number of houses. For special events, the variables yz appears as two dots. The dots on the second line are replaced by the player letters that are currently on this cell. All four players could be in the same cell; therefore, we have four slots. If there are no players on a cell, all four slots appear as dot characters. For example, players t and n are on the property cell C in the map sample above. Property C is owned by the player t and has one house on it.

**Preliminary Submission:** You will submit an early version of your solution before the final submission. This version should at least include the following:

- Turn taking, player movement and special events should be completed.
- You should print the map and each player's coins between turns.

You will have time to complete your solution after you submit your preliminary solution. You can consult the TAs and tutors during the lab. Do not forget to make your final submission at the end. Even if you finish the assignment in the preliminary submission, you should submit for the final submission on Moodle.

**Not completing the preliminary submission on time results in 50% reduction of this assignment's final grade.**