**CSE1341 – Lab #5**

*First version of game with one class and multiple methods*

# PRE-LAB [10 points]

**Must be done prior to your lab session.**

Create the class named SMUQuest.java with the shells for the main, rollDie, getMoneyAmount, and getDestination methods. Read the instructions carefully and make sure you have the proper method headers for each of those methods (including parameters and return type) based on the information provided in the instructions below.

# LAB [90 points]

NOTES: Use the given notes as a guide for the program logic. These comments must be included in the programs to explain the logic followed.

Each program should compile without errors and should run to produce outputs described for each exercise. The following points will be discounted if the related element is missing or incorrect:

Output formatting matches the output in the instructions [20 points] Proper names for classes and variables [15 points]

Proper method headers with parameters and return type [5 points each] Program logic/algorithms/steps work as described in the instructions [20 points] Comments [15 points]

Program doesn't compile [ 20 points] Source code (java file) missing [ 15 points] Executable (class file) missing [15 points]

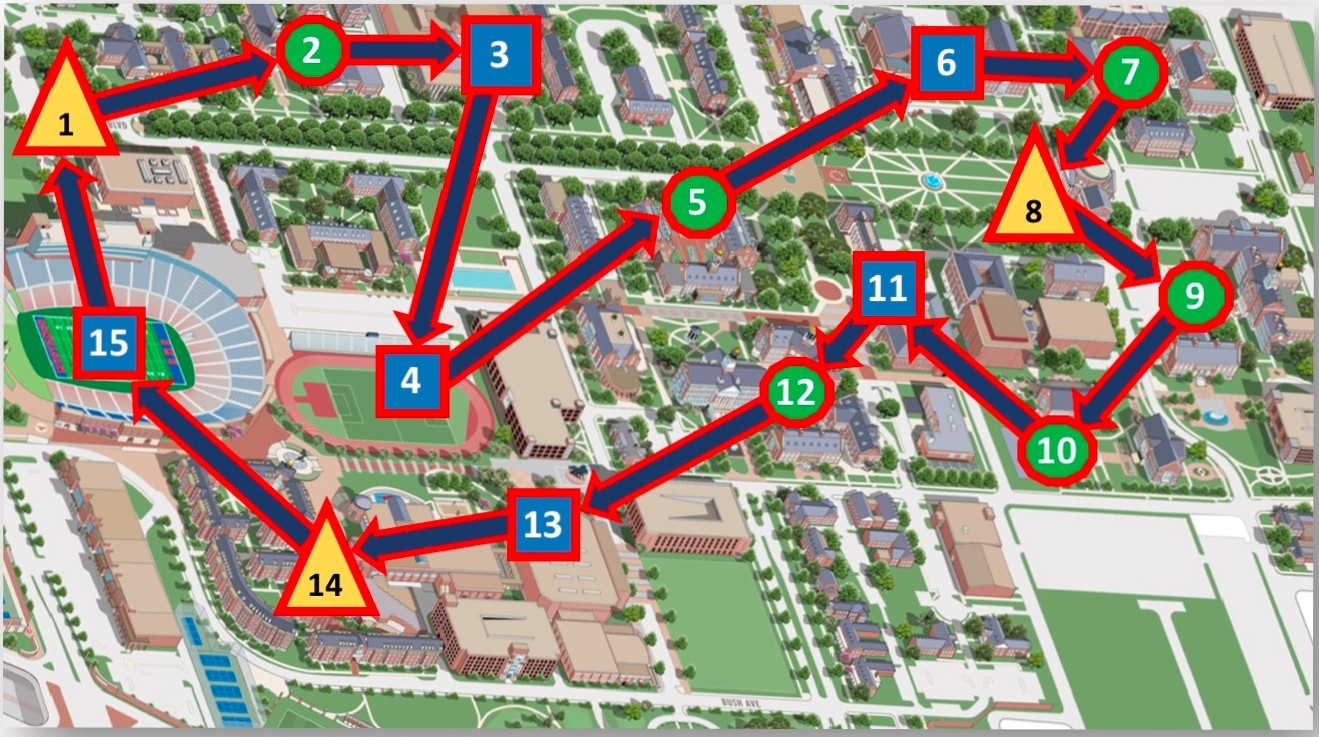
*Instructions:*

Create the SMUQuest game as a single class with three methods described in the instructions. When starting the game, the user will be prompted for the names of each of the two players, as well as the dollar amount they must attain to win the game.

The remainder of the game is a simulation of two players alternating turns until one of them wins the game. There is no user input…just show the output of continuous play until a winner is declared.

Make sure the format of your output matches the sample output provided in the instructions. Of course, your actual output will be different based on the actual simulated die rolls.

**SMUQuest**





**9** Collect $200

IIPay $100

*Even die roll:*

Collect 10 times die amount

*Odd die roll:*

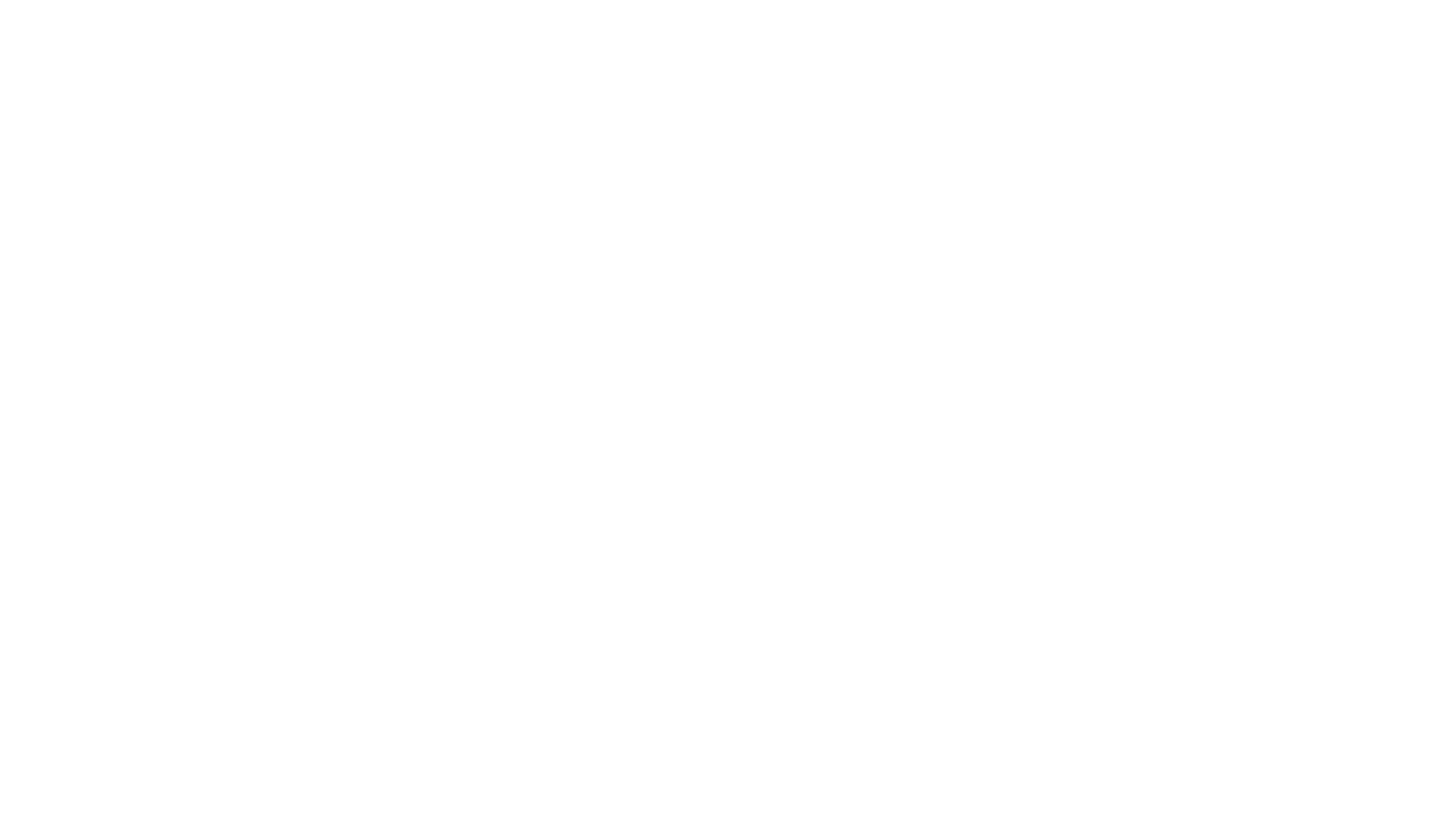
Pay 10 times die amount

**Game Play**

1. Admire the campus view down *Bishop Boulevard*
2. Visit *Perkins Theology* School
3. Attend a concert at *Meadows*
4. Watch a Soccer game at *Westcott Field*
5. Take a class *at Cox School of Business*
6. Watch a show at *McFarlin* Auditorium
7. Study at *Dedman Law Library*
8. See the downtown skyline from *Dallas Hall*
9. Do an experiment at *Fondren Science*
10. Take a class at *Simmons*
11. Get a snack at *Hughes-Trigg Student Center*
12. Learn Java programming at *Lyle*
13. Watch a basketball game at *Moody*
14. Take a nap at *residential commons*
15. Cheer for Mustang Football at *Ford Stadium*

Play begins at Bishop Boulevard and each player has $500. Each player takes a turn rolling one die and moving forward to that location. The new location is displayed along with the player name, money adjustments and new balance. Play continues to rotate between players. The board is a continuous loop...at Ford Stadium, play continues back at Bishop Boulevard. The first player to have a total of $5000 or more is the winner.

**Class:** SMUQuest



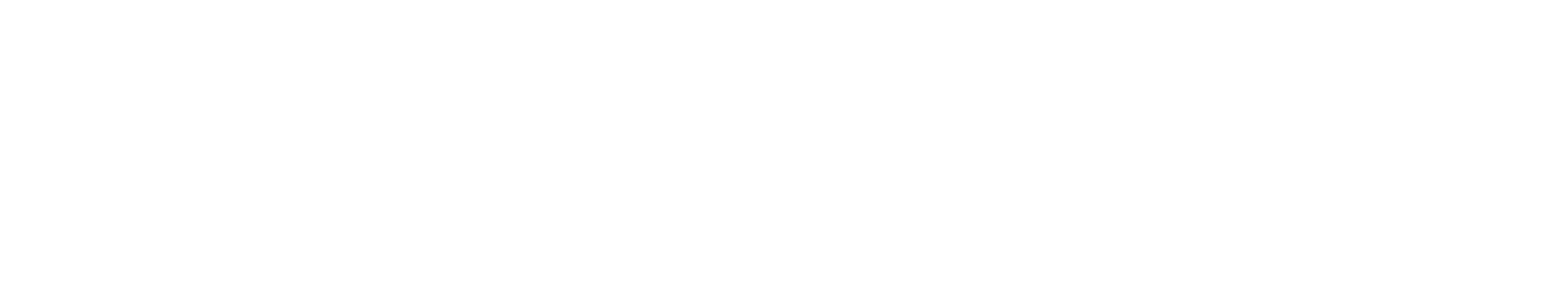
**Method:** *main*

**Returns**: *void*

**Parameters**: String[] (Standard main method signature)

**Logic:**

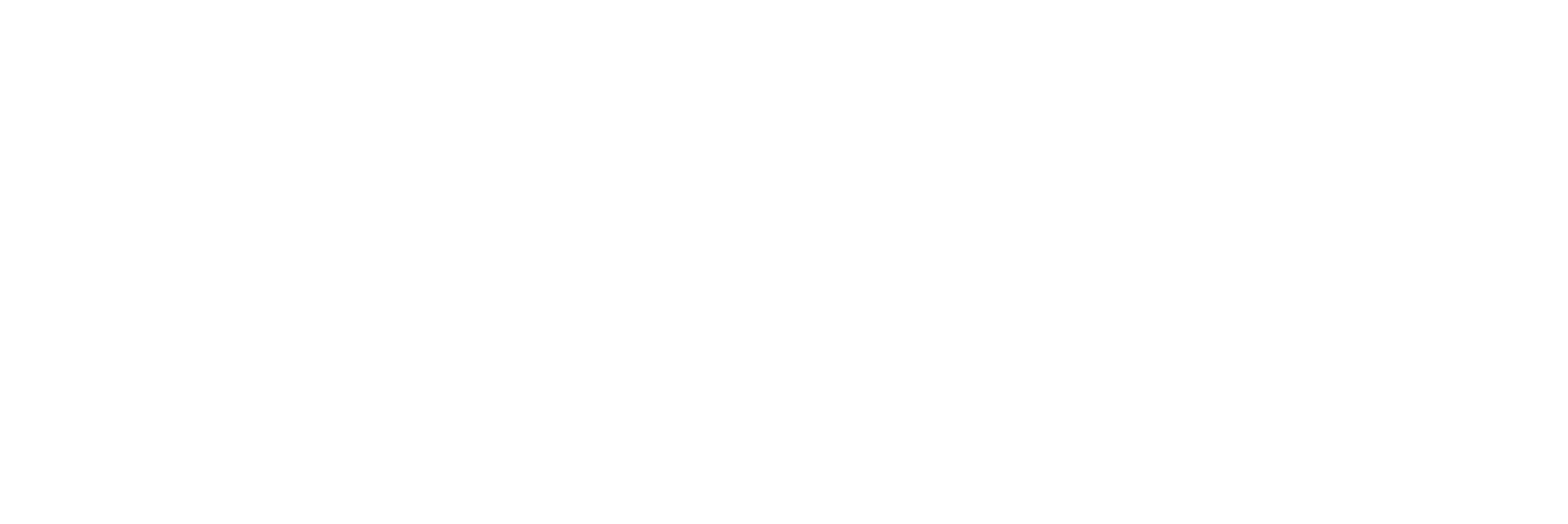
1. Store the 15 location descriptions from the instructions in an array
2. Display welcome message
3. Prompt for player 1 name and set that player’s starting amount to 500
4. Repeat for player 2
5. Prompt for the money amount that must be attained to win the game
6. Loop until one of the player reaches or exceeds the winning amount
   1. Call the rollDie method
   2. Call the getDestination method and set the player’s location to the new location it returns
   3. Call the getMoneyAmount method to determine how much to add or subtract from the player’s money
   4. Print the player’s new location and money amount following the format shown in the sample output
   5. Repeat these steps for the second player
   6. Check each player’s money amount and break out of the loop when one reaches or passes the winning value
7. Display the winner and exit the game



**Method:** *rollDie* **Returns**: *int* **Parameters**: n/a

**Logic:**

1. Generate a random number in the range of 1..6 (inclusive)
2. Return that value



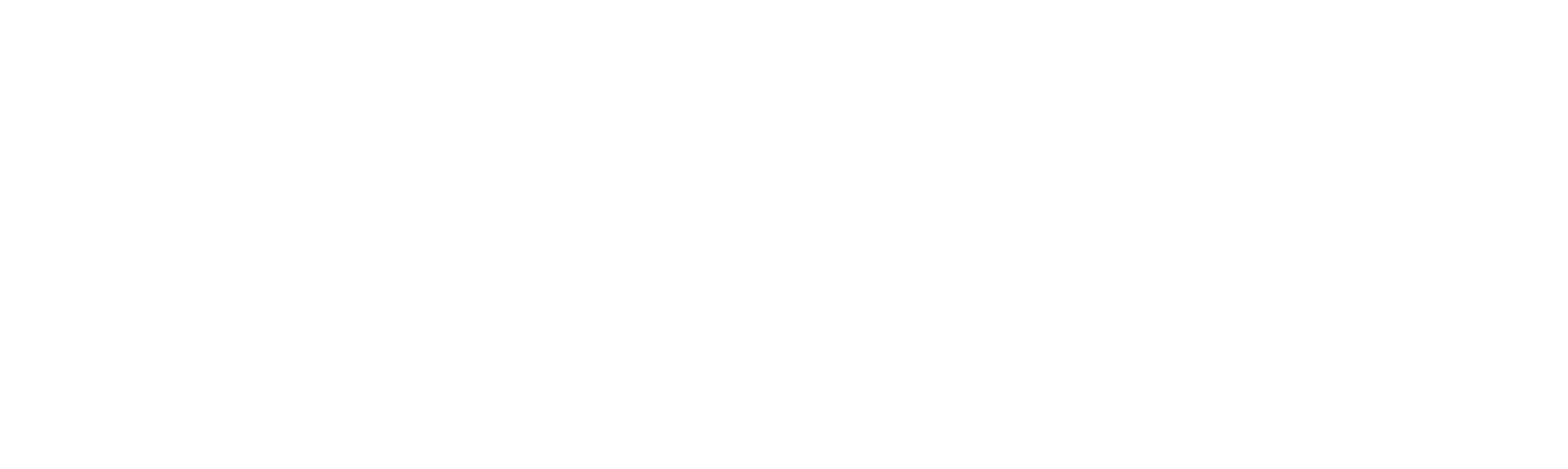
**Method:** *getMoneyAmount*

**Returns**: *int (amount to be added or subtracted from the player’s money in the main method*

**Parameters**: space number (int) and roll value (int)

**Logic:**

1. Create an array containing the money values of the 15 game locations. They should either be 200, -100 or 0.
2. If the location has 0, determine if the roll value passed in is odd or even
   1. If the roll value is even, set the amount you will return to ten times the roll value
   2. If it’s odd, set the amount you will return to zero minus ten times the roll value
3. Return the location value (200, -100 or the amount you calculated in step 2



**Method:** *getDestination*

**Returns**: *int (new playing board location that the player will be moved to in the main method)*

**Parameters**: current space number (int) and roll value (int)

**Logic:**

1. Create an algorithm that determines the player’s new location by moving around the board the number of spaces indicated by the roll value passed in. Remember that the playing field has 15 locations (index locations 0 to 14) and that it’s a continuous loop – when the player reaches the last space, they continue to move around the board from the first space.
2. Return the new playing board location to the main method.

## Sample Output:

$ **java SMUQuest** Welcome to SMUQuest! Player 1 name: **Peruna** Peruna has $500 Player 2 name: **Bevo** Bevo has $500

How much is needed to win? **1000**

Peruna rolled 3 and stopped to watch a Soccer game at Westcott Field.

Peruna lost $100 and now has $400

Bevo rolled 2 and stopped to attend a concert at Meadows.

Bevo lost $100 and now has $400

Peruna rolled 6 and stopped to take a class at Simmons.

Peruna collected $200 and now has $600

Bevo rolled 3 and stopped to watch a show at McFarlin Auditorium.

Bevo lost $100 and now has $300

Peruna rolled 4 and stopped to take a nap at residential commons.

Peruna collected $40 and now has $640

Bevo rolled 3 and stopped to do an experiment at Fondren Science.

Bevo collected $200 and now has $500

Peruna rolled 3 and stopped to visit Perkins Theology School.

Peruna collected $200 and now has $840

Bevo rolled 1 and stopped to take a class at Simmons.

Bevo collected $200 and now has $700

Peruna rolled 3 and stopped to take a class at Cox School of Business.

Peruna collected $200 and now has $1040

Bevo rolled 4 and stopped to take a nap at residential commons.

Bevo collected $40 and now has $740 Peruna won! GAME OVER!!

## Submit the *java* and *class* files via Canvas (as a single zip-file). Include a comment block at the top of each *Java* file that includes your name, student id number, and “Lab 5-Fall 2017”.

**This assignment is due by 6:00AM Saturday, October 28.**