CS 312 – Final – Fall 2022 SOLUTIONS

Part 1 - Expressions - 10 questions, 2 points each, 20 points total

```
A: 6.8
B: true
C: false
D: “9UT57”
E: ‘N’
F: false
G: ‘F’
H: false
I: 1.5
J: 5
```

Part 2 - Code Tracing - 12 questions, 4 points each, 48 points total

```
A:   true false
B:   8 4 5 3
C:   0 5
D:   6.0 true
E:   [-6, 4, -1, 7, 11]
F:   THERE IS NO QUESTION F
G:   [-6, -12, -26, -92]
H:   runtime error
I:   10 40
J:   [11, -4, 10]
K:   [P, P, X, P, K, X]
L:   225
M:   ->*
```

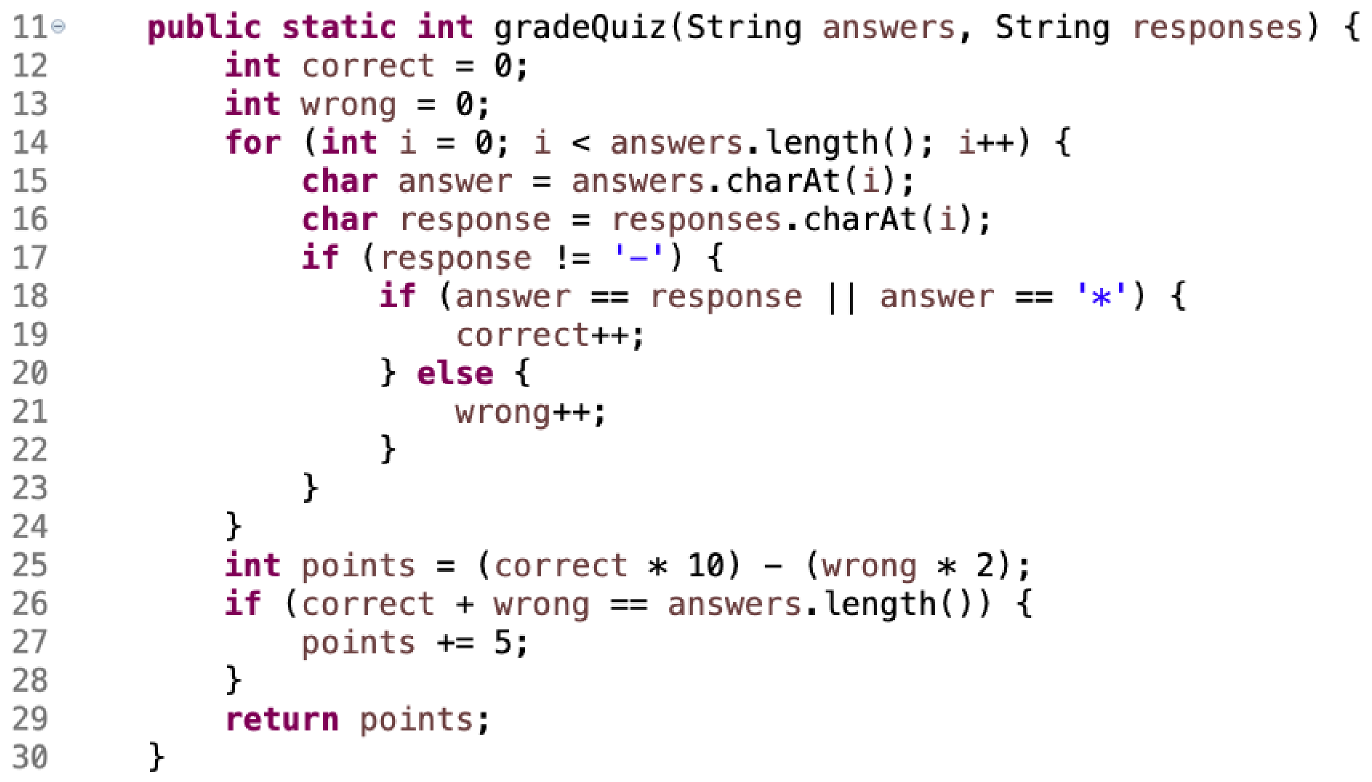
Part 3 - Short Answer (OOP) - 3 questions, 2 points each, 6 points total

A. syntax error, legal, syntax error, syntax error

B.
```
Type: CIRCLE
Color: java.awt.Color[r=0,g=0,b=255]
Radius: 3.0
Area: 28.3
Perimeter: 18.8
```

C. The Shape class doesn’t have a constructor that takes no arguments. Also, since a constructor is provided, the default constructor is no longer available.

Part 4A - Programming - gradeQuiz() method (Chapters 1-6) - 26 Points



```
11    public static int gradeQuiz(String answers, String responses) {
12        int correct = 0;
13        int wrong = 0;
14        for (int i = 0; i < answers.length(); i++) {
15            char answer = answers.charAt(i);
16            char response = responses.charAt(i);
17            if (response != '-') {
18                if (answer == response || answer == '*') {
19                    correct++;
20                } else {
21                    wrong++;
22                }
23            }
24        }
25        int points = (correct * 10) - (wrong * 2);
26        if (correct + wrong == answers.length()) {
27            points += 5;
28        }
29        return points;
30    }
```

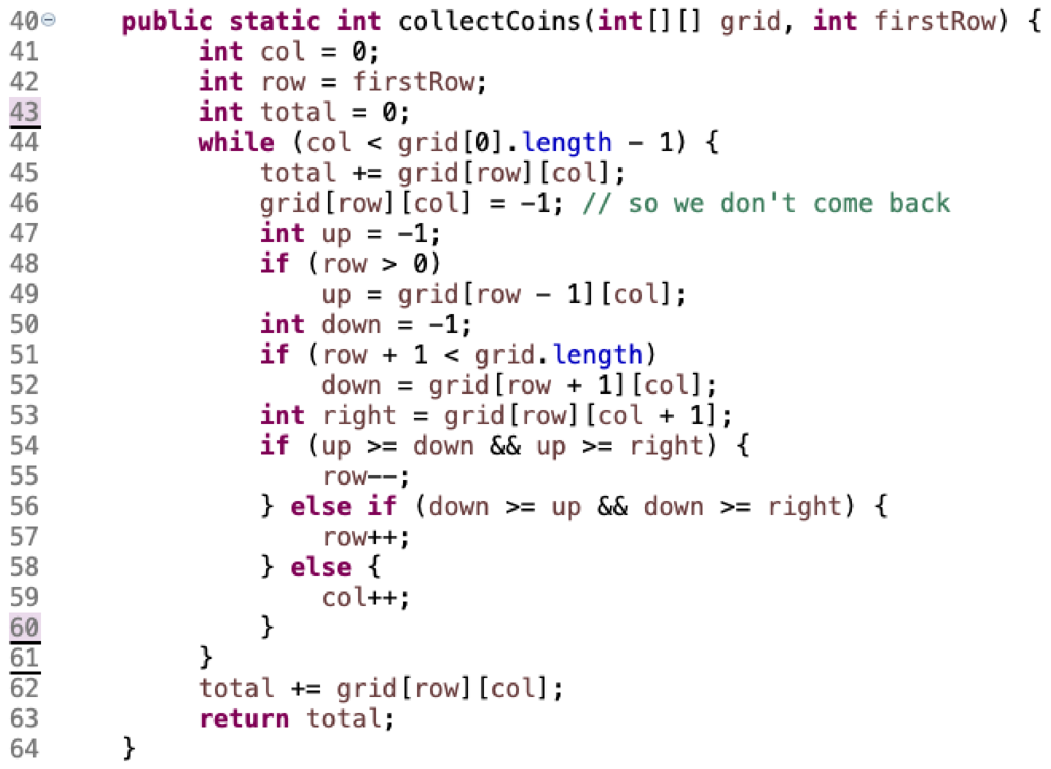
Item	Line #	Item	Points
A	11	Method declaration	+2 int return type +2 two String parameters
B	14	For Loop	+2 correct array traversal for loop, 0 - < length +2 correct access to ith item in both strings
C	19 - 25	Score calculation	+2 correct handling of - in studentAnswers (not right or wrong and no credit for * correctAnswer) +2 correct handling of * in correctAnswers (if student responded then any answer is correct) +2 correct adding of 5 points if all questions answered +2 correct math for final score
D	20	Return	+2 correct return of final score
E	N/A	GENERAL	+2 No unnecessary code +2 No redundant or inefficient code +2 No Java syntax not allowed +2 No seriously incorrect style guide issue (if line count == 0, 0 max points, else if line count < 6, max 4 points, else max 8 points)

Part 4B - Programming - capitalLettersFreqPrint() method - 1D Arrays - 32 Points

```
16      public static int capitalLettersFreqPrint(Scanner sc) {
17          int[] freqs = new int[NUM_CAPS];
18          while (sc.hasNext()) {
19              String word = sc.next();
20              for (int i = 0; i < word.length(); i++) {
21                  char ch = word.charAt(i);
22                  if ('A' <= ch && ch <= 'Z') {
23                      int index = ch - 'A';
24                      freqs[index]++;
25                  }
26               }
27           }
28          System.out.println ("The non-zero frequencies are: ");
29          int total = 0;
30          for (int i = 0; i< NUM_CAPS; i++) {
31              if (freqs[i] != 0) {
32                  System.out.println ((char)(i + 'A') + " - " + freqs[i]);
33                  total += freqs[i];
34              }
35          }
36          return total;
37      }
```

Item	Line #	Item	Points
A	15	Method declaration	+2 correct Scanner parameter +2 correct int return type
B	17	array	+1 correct declaration for array using NUM_CAPS constant
C	18	while loop	+1 correct loop test, while hasNext() +1 correct reading of next token using next()
D	20	for loop	+1 correct traversal loop +2 correct access to ith character in token
E	22-24	tallies	+2 if statement to test for a capital letter +2 correct array index +2 correct increment of correct element of frequency array
F	28-35	print	+2 correct for loop to traverse frequency array +2 correct test to print only non-zero frequencies +2 correct print of the letter, - , frequency +1 correct addition to total
G	36	return	+1 correct return of the total
H	N/A	GENERAL	+2 No unnecessary code +2 No redundant or inefficient code +2 No Java syntax not allowed +2 No seriously incorrect style guide issue (if line count == 0, 0 max points, else if line count < 6, max 4 points, else max 8 points)

Part 4C - Programming - collectCoins() - 2D Arrays - 34 Points



```
40    public static int collectCoins(int[][] grid, int firstRow) {
41        int col = 0;
42        int row = firstRow;
43        int total = 0;
44        while (col < grid[0].length - 1) {
45            total += grid[row][col];
46            grid[row][col] = -1; // so we don't come back
47            int up = -1;
48            if (row > 0)
49                up = grid[row - 1][col];
50            int down = -1;
51            if (row + 1 < grid.length)
52                down = grid[row + 1][col];
53            int right = grid[row][col + 1];
54            if (up >= down && up >= right) {
55                row--;
56            } else if (down >= up && down >= right) {
57                row++;
58            } else {
59                col++;
60            }
61        }
62        total += grid[row][col];
63        return total;
64    }
```

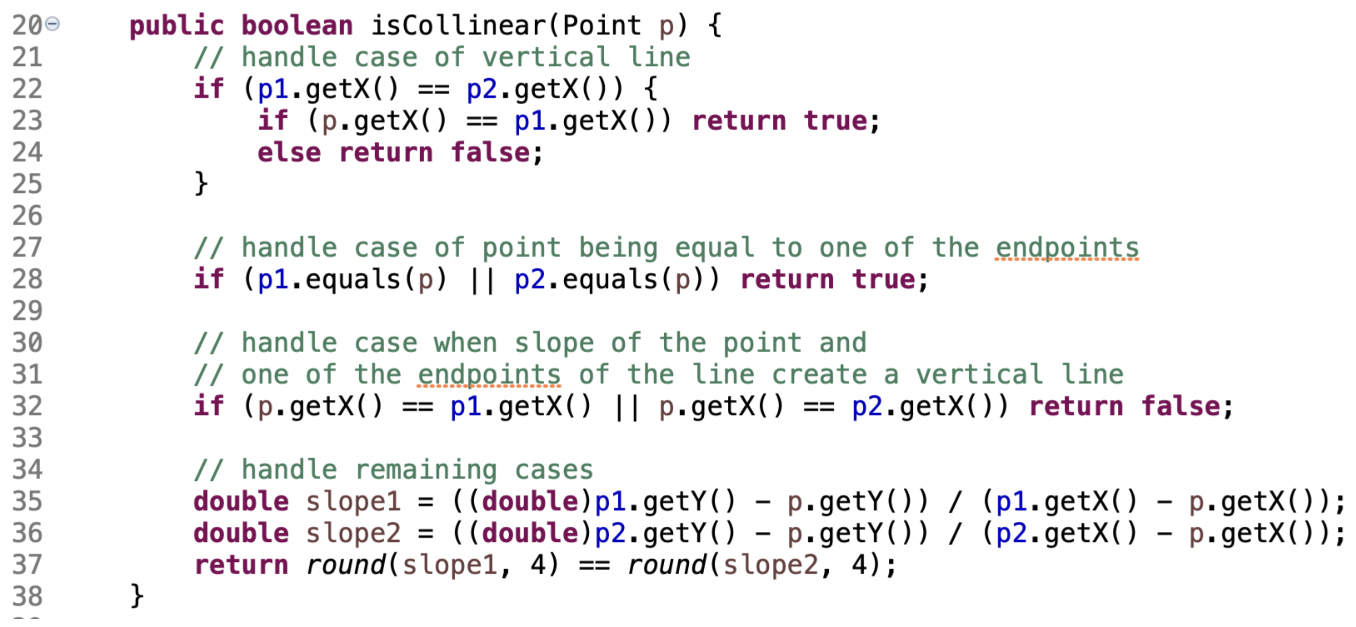
Item	Line #	Item	Points
A	15	Method declaration	+2 correct int[][] grid and int row parameters +2 correct int return type
B	41-43	initialization	+2 correct initialization for row, col and total
C	44	while loop and total	+2 continue until robot has reached last column +2 correct increment for total inside while +2 correct increment for total for last cell outside while
D	46-60	movement	+4 correct mechanism to indicate cells have been visited +4 correct mechanism to get 3 adjacent coin counts +4 correct decision and increment for up, down or right
E	63	return	+2 correct return of total
F	N/A	GENERAL	+2 No unnecessary code +2 No redundant or inefficient code +2 No Java syntax not allowed +2 No seriously incorrect style guide issue (if line count == 0, 0 max points, else if line count < 6, max 4 points, else max 8 points)

Part 4D - Programming - Sloth - OOP - 38 Points

```java
import java.awt.Color;

public class Sloth extends Critter {

    private static final int MOVE_WAIT = 30;
    private static final int EAT_WAIT = 5;
    private static final Direction[] SLOTH_DIRECTIONS = {Direction.NORTH,
            Direction.EAST, Direction.SOUTH, Direction.WEST};

    private int numMoveCalls;
    private int numMoves;
    private int numEatCalls;

    public Sloth () {
        numMoveCalls = 0;
        numMoves = 0;
        numEatCalls = 0;
    }

    public Attack fight (String opponent) {
        if (opponent.equals("%")) return Attack.FORFEIT;
        else return Attack.SCRATCH;
    }

    public Color getColor () { return Color.GREEN; }

    public Direction getMove() {
        numMoveCalls++;
        Direction direction = Direction.CENTER;
        if (numMoveCalls %  MOVE_WAIT == 0) {// time to move
            numMoves++;
            direction = SLOTH_DIRECTIONS[numMoves % SLOTH_DIRECTIONS.length];
        }
        return (direction);
    }

    public boolean eat () {
        numEatCalls++;
        if (numEatCalls % EAT_WAIT == 0) return true;
        else return false;
    }

    public String toString() { return ("0"); }

}
```

Item	Line #	Item	Points
A	3	Class header	+2 correct class header and name +2 correct extends Critter
B	5-7	Constants	+2 class constants for MOVE_WAIT, EAT_WAIT and SLOTH_DIRECTIONS
C	10-12	fields	+2 private fields for NumMoveCalls, numMoves, numEatCalls or similar
D	14-18	constructor	+2 correct field initialization
E	20-22	fight() method	+2 correct Attack return type and String parameter type +2 correct return of either FORFEIT or SCRATCH
F	25	getColor() method	+2 correct method, no parameters and Color.GREEN return
G	27-35	getMove() method	+4 correct decision to move or not move +4 correct choice for direction to move and return
H	37-41	eat() method	+2 correct decision to eat or not, based on eat count +2 correct return of decision
I	43	toString() method	+2 correct return of "O"
J	N/A	GENERAL	+2 No unnecessary code +2 No redundant or inefficient code +2 No Java syntax not allowed +2 No seriously incorrect style guide issue (if line count == 0, 0 max points, else if line count < 6, max 4 points, else max 8 points)

Part 4E - Programming - isCollinear() method - OOP - 36 Points



```
20      public boolean isCollinear(Point p) {
21          // handle case of vertical line
22          if (p1.getX() == p2.getX()) {
23              if (p.getX() == p1.getX()) return true;
24              else return false;
25          }
26
27          // handle case of point being equal to one of the endpoints
28          if (p1.equals(p) || p2.equals(p)) return true;
29
30          // handle case when slope of the point and
31          // one of the endpoints of the line create a vertical line
32          if (p.getX() == p1.getX() || p.getX() == p2.getX()) return false;
33
34          // handle remaining cases
35          double slope1 = ((double)p1.getY() - p.getY()) / (p1.getX() - p.getX());
36          double slope2 = ((double)p2.getY() - p.getY()) / (p2.getX() - p.getX());
37          return round(slope1, 4) == round(slope2, 4);
38      }
```

Item	Line #	Item	Points
A	20	Method declaration	+2 Point parameter +2 boolean return type
B	22-25	Vertical line	+2 check for a vertical line, +2 if point is collinear, returns true, else returns false
C	28	Point equal to endpoint	+2 correct test using .equals() from Point class
D	32	Undefined slope between point and endpoints	+2 check for potential undefined slope +2 return false (only case when it would be true has already been handled)
E	35-37	Slope calculations	+4 correct slope for line P1 and the new point with cast to double +4 correct slope for line P2 and the new point with cast to double +4 uses round() before slope comparison
F	37	return	+2 correct return
G	N/A	GENERAL	+2 No unnecessary code +2 No redundant or inefficient code +2 No Java syntax not allowed +2 No seriously incorrect style guide issue (if line count == 0, 0 max points, else if line count < 6, max 4 points, else max 8 points)