

Introduction to Programming (Java)

Example solutions – Exam December 2018

Q1 (a) 1 2
1 2
11 2
11 18
17 18
17 18

(b)

```
double calculate(double x, int n) {
    double result = 0;
    int sign = 1;
    int faculty = 1;
    double power = x * x;

    for (int i = 1; i <= n; i++) {
        faculty *= i;
        power *= x;
        double term = sign * (faculty / ((i + 1) * power));
        result += term;
        sign *= -1;
    }

    return result;
}
```

(c)

```
static final int NUMBER_OF_ROWS = 8,
               NUMBER_OF_COLUMNS = 4;

char[][] matrix = new char[NUMBER_OF_ROWS][NUMBER_OF_COLUMNS];

int numberOfTops(int[][] m) {
    int result = 0;

    for (int row = 1; row < m.length - 1; row++) {
        for (int column = 1; column < m[0].length - 1; column++) {
            if (m[row][column] > m[row - 1][column] &&
                m[row][column] > m[row + 1][column] &&
                m[row][column] > m[row][column - 1] &&
                m[row][column] > m[row][column + 1]) {
                result += 1;
            }
        }
    }

    return result;
}
```

(d) 5 6
16 7
7 9
22 10
7 22
10 30

Q2 (a) class CoinCollection {
 static final int MAX_NUMBER_OF_ELEMENTS = 2500;

 Coin[] elements;
 int numberOfElements;

 CoinCollection() {
 elements = new Coin[MAX_NUMBER_OF_ELEMENTS];
 numberOfElements = 0;
 }

 void add(Coin coin) {
 elements[numberOfElements] = coin;
 numberOfElements += 1;
 }
}

(b) Add to the class Coin:

```
static final int RARE_VALUE_BORDER = 10;

boolean isRare() {
    return grade.equals("mint state") && printedValue > RARE_VALUE_BORDER;
}
```

Add to the class CoinCollection:

```
CoinCollection rareCoins() {
    CoinCollection result = new CoinCollection();

    for (int i = 0; i < numberOfElements; i++) {
        if (elements[i].isRare()) {
            result.add(elements[i]);
        }
    }

    return result;
}
```

(c) Add to the class Coin:

```
static final int MODERN_MINTED_BORDER = 1945;

boolean isModern() {
    return yearMinted > MODERN_MINTED_BORDER;
}
```

Add to the class CoinCollection:

```
void removeModernCoins() {
    for (int i = 0; i < numberOfElements; i++) {
        if (elements[i].isModern()) {
            elements[i] = elements[numberOfElements - 1];
            numberOfElements -= 1;
            i -= 1;
        }
    }
}
```

Q2 (d) Add to the class CoinCollection:

```
int rareOldCoinsFrom(String origin) {
    CoinCollection rare = rareCoins();
    rare.removeModernCoins();
    return rare.coinsFrom(origin);
}
```

Q3 (a) `int count(int[] r, int i, int a) {`

```
    if (i == r.length) {
        return 0;
    }

    if (r[i] == a) {
        return 1 + count(r, i + 1, a);
    } else {
        return count(r, i + 1, a);
    }
}
```

(b) `int count(String s, String pattern) {`
 `if (s.length() < pattern.length()) {`
 `return 0;`
 `}`

```
    return (s.substring(0, pattern.length()).equals(pattern) ? 1 : 0) +
           count(s.substring(1), pattern);
}
```