OBJECT-ORIENTED PROGRAMMING(CPSC 1811)

Assignment #3

Due Date: November 20, 2022

Mourad Bouguerra
mbouguerra@langara.ca

Part-I
Inheritance: Extending an Abstract Class

(75 marks)

❒ Given the following startup code of a **Java** classes

Listing 1: Fibonacci Class

```
1  package fibonacci;
2  import java.math.BigInteger;
3  import java.util.HashMap;
4
5  public abstract class Fibonacci {
6    public BigInteger usingRecursion(int n)
7    {
8      BigInteger fibonacci = BigInteger.valueOf(n);
9      if(n <= 1) {
10       return fibonacci;
11     }
12     return usingRecursion(n-1).add(usingRecursion(n-2));
13   }
14   public abstract BigInteger usingIteration(int n);
15   public abstract BigInteger usingTailRecursion(int n, BigInteger a, BigInteger b);
16   public abstract BigInteger usingMemoization(int n,HashMap<Integer,BigInteger> cache);
17
18 }
```

Listing 2: FibonacciComputation Class

```
1  package fibonacci;
2
3  import java.math.BigInteger;
4  import java.util.HashMap;
5
6  public class FibonacciComputation extends Fibonacci {
7    @Override
8    public BigInteger usingIteration(int n) {
9      // your code goes here
10   }
11   @Override
12   public BigInteger usingTailRecursion(int n, BigInteger a, BigInteger b) {
13     // your code goes here
14   }
15   @Override
16   public BigInteger usingMemoization(int n, HashMap<Integer,BigInteger> cache) {
17     // your code goes here
18 }
```

1. Fibonacci Computation **(85 marks)**

(a) Write the definition of the **BigInteger usingIteration(int n)** method to compute `Fibonacci` number using iteration **(25 marks)**

(b) Write the definition of the **BigInteger usingTailRecursion(int n)** method to compute `Fibonacci` number using **tail recursion** **(30 marks)**

(c) Write the definition of the **BigInteger usingMemoization(int n)** method to compute `Fibonacci` number using **memoization** **(30 marks)**

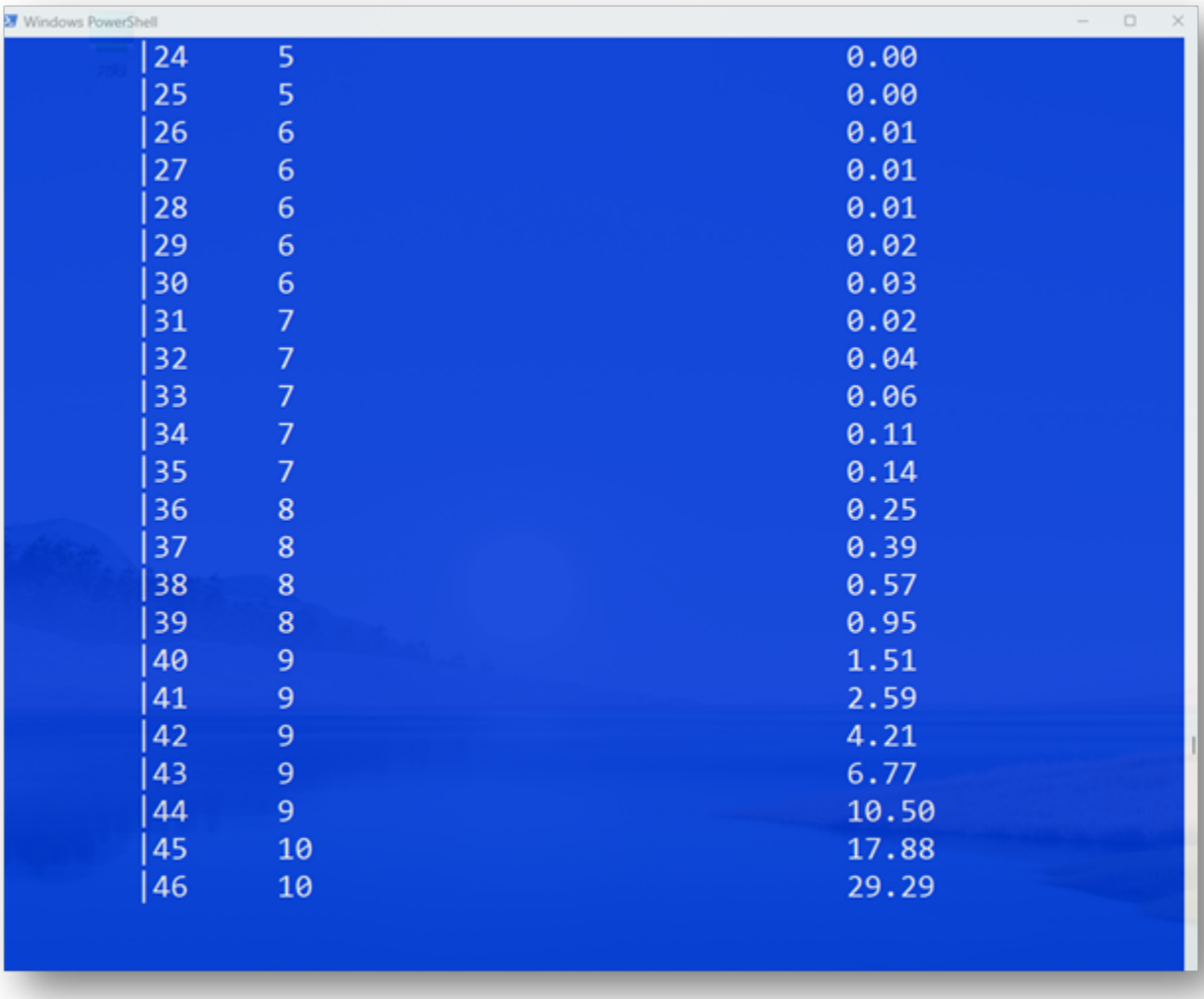

Figure 1: Using Recursion Output

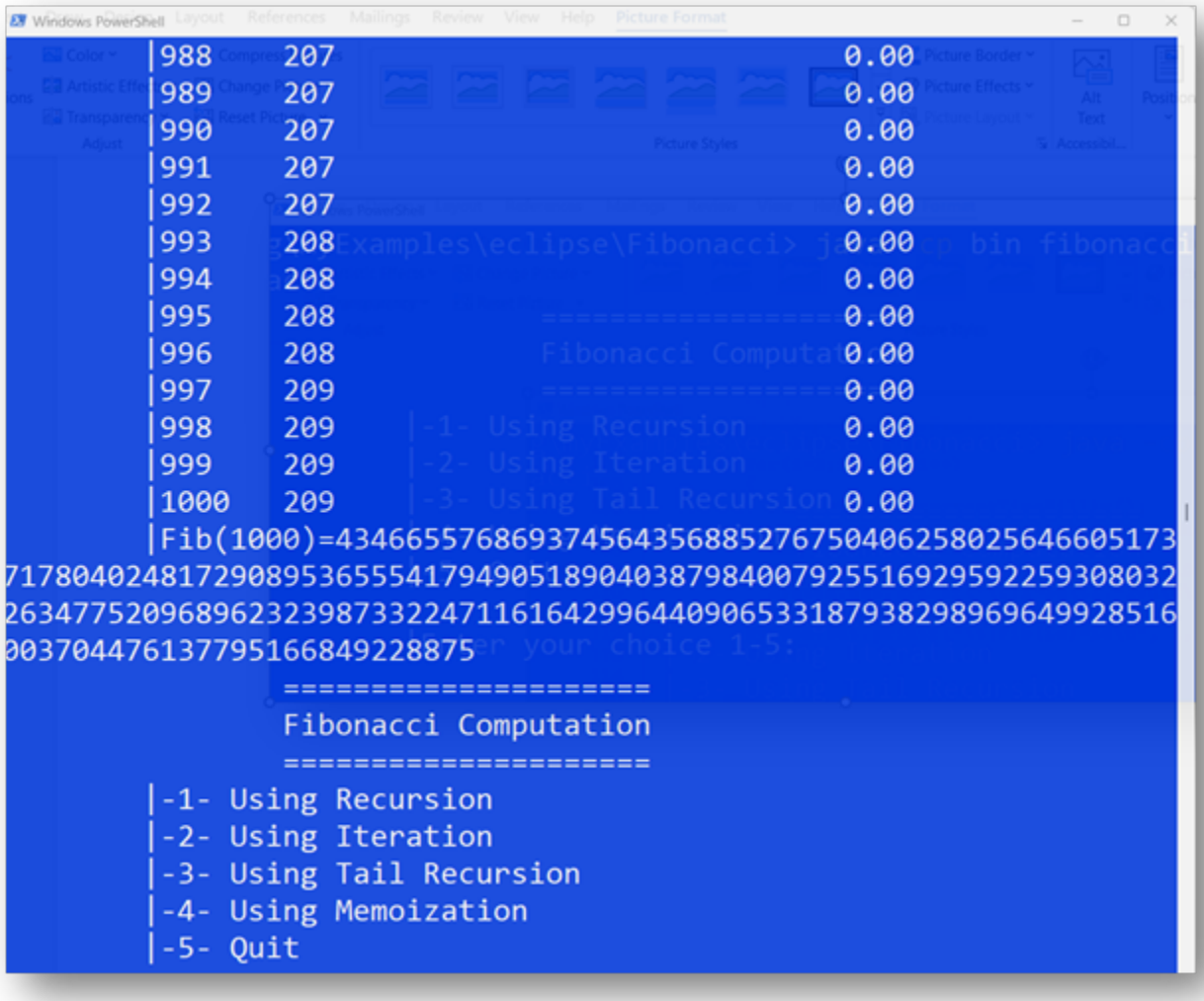

Figure 2: Using Iteration/Tail Recursion/Memoization Output

2. Use the following class to test your `FibonacciComputation` class (see figures 1 & 2 on pages 4 & 5)

Listing 3: TestingFibonacci Class

```
1  package fibonacci;
2  import java.util.HashMap;
3  import java.util.InputMismatchException;
4  import java.util.Scanner;
5  import java.math.BigInteger;
6
7  public class TestingFibonacci {
8    final static int QUIT=5;
9      public static void displayMenu() {
10         System.out.println("\t\t====================");
11         System.out.println("\t\tFibonacci Computation");
12         System.out.println("\t\t====================");
13         System.out.println("\t|-1- Using Recursion");
14         System.out.println("\t|-2- Using Iteration");
15         System.out.println("\t|-3- Using Tail Recursion");
16         System.out.println("\t|-4- Using Memoization");
17         System.out.println("\t|-5- Quit\n");
18         System.out.print("\t|Enter your choice 1-5:  ");
19     }
20     public static int getMenuChoice() {
21       Scanner keyboard = new Scanner(System.in);
22       int choice = 0;
23       try {
24         //get input from the keyboard
25         choice = keyboard.nextInt();
26         // System.out.println("x = " + x);      //debug
27         while ((1 > choice) || (choice > QUIT)) {
28           System.out.println("\n\tError: Invalid entry.");
29           System.out.print("\n\tEnter the number corresponding to the menu entry (1-9): ");
30           choice = keyboard.nextInt();
31         }
32       }
33       catch (InputMismatchException ex) {
34         //The input type was wrong (eg. a letter).
35         System.err.println("\t|You must enter an integer between 1 and 5.");
36         System.exit(1);
37       }
38       catch (Exception ex) {      //all other exceptions
39         System.err.println("\t|Something went wrong!" );
40         System.exit(1);
41
42       }
43       return choice;
44     }
45     public static void displayHeader() {
46       System.out.println("\t============================");
47     System.out.println("\t|Computing Fibonacci Sequence\n");
48     System.out.println("\t============================");
49     System.out.println("\t|N\t# of Digits of Fib(N)\t\tTime to Compute (Seconds)");
50     System.out.println("\t==\t====================\t\t========================");
51     }
52
53   public static void main(String[] args) {
54     int choice = 1;
55     int i;
56     long start,end;
57     BigInteger x = BigInteger.valueOf(-1);
58     Fibonacci fibonacci = new FibonacciComputation();
59     do
60     {
61       displayMenu();
62       choice = getMenuChoice();
63       switch(choice)
64       {
65       case 1:
66         displayHeader();
67         for(i=2;i<1001;i++)
68         {
69           start=System.currentTimeMillis();
70            x = fibonacci.usingRecursion(i);
71           end = System.currentTimeMillis();
72           float duration =(float)(end-start)/1000;
73           System.out.printf("\t|%-3d\t%,-10d\t\t\t%.2f\t\n",i,x.toString().length(),duration);
74         }
75         System.out.printf("\t|Fib(%d)=%d\n", i-1,x);
76         break;
77       case 2:
78         displayHeader();
79         for(i=2;i<1001;i++)
80         {
81           start=System.currentTimeMillis();
82           x = fibonacci.usingIteration(i);
83           end = System.currentTimeMillis();
84           float duration =(float)(end-start)/1000;
85           System.out.printf("\t|%-3d\t%,-10d\t\t\t%.2f\t\n",i,x.toString().length(),duration);
```

```
86          }
87          System.out.printf("\t|Fib(%d)=%d\n", i-1,x);
88          break;
89        case 3:
90          displayHeader();
91          for(i=2;i<1001;i++)
92          {
93            start=System.currentTimeMillis();
94            x = fibonacci.usingTailRecursion(i,BigInteger.ZERO,BigInteger.ONE);
95            end = System.currentTimeMillis();
96            float duration =(float)(end-start)/1000;
97            System.out.printf("\t|%-3d\t%,-10d\t\t\t%.2f\t\n",i,x.toString().length(),duration);
98          }
99          System.out.printf("\t|Fib(%d)=%d\n", i-1,x);
100         break;
101       case 4:
102         HashMap<Integer,BigInteger> cache = new HashMap<Integer,BigInteger>();
103         displayHeader();
104         for(i=2;i<1001;i++)
105         {
106           start=System.currentTimeMillis();
107           x = fibonacci.usingMemoization(i,cache);
108           end = System.currentTimeMillis();
109           float duration =(float)(end-start)/1000;
110           System.out.printf("\t|%-3d\t%,-20d\t\t\t%.2f\t\n",i,x.toString().length(),duration);
111         }
112         System.out.printf("\t|Fib(%d)=%d\n", i,x);
113         break;
114
115       case 5:
116         System.out.println("\t|You chose to quit!");
117         System.exit(0);
118         break;
119       default:
120         System.out.println("\t|You should not be here!");
121       }
122     }while (choice != QUIT);
123
124   }
125
126 }
```

Submission

✎ You have to submit a `FibonacciComputation.java` file on the Brightspace submission page.

Marking Scheme

Task	Marks
`Using Iteration`	25
`Using Tail Recursion`	30
`Using Memoization`	30
`Coding Style`	15
Total	**100**