

CS20B: Homework 1 (100 points)

prof. dr. Irma Ravkic

Fall, 2019

1 Questions (55 points)

1. (2 points) What is the difference between an object and a class? Give two examples.

2. (2 points) Explain the difference between

```
1  int a = 3;
```

and

```
1  Integer a = new Integer(3);
```

3. (10 points) For the following concepts/classes define/write at least one method:

- (a) Animal
- (b) Dog
- (c) User
- (d) File
- (e) Database

4. (10 points) For the following concepts/classes write at least one instance variable/attribute and an associated constructor you can define taking **more than one input argument/parameter**:

- (a) MobilePhone
- (b) User
- (c) File
- (d) Database
- (e) Webpage

5. (2 points) What is the output of the following program:

```
1  class T {  
    private int t = 20;  
3    T(){  
        t = 40;  
5    }  
}
```

```

    }
7   class Main {
        public static void main(String args[]) {
9           T t1 = new T();
            System.out.println(t1.t);
11        }
    }

```

6. (2 points) Is there any compiler error in the below Java program? Please explain your answer. I don't accept only yes or no.

```

class Point {
2   int m_x, m_y;
    public Point(int x, int y) {m_x = x;m_y = y;}
4   public static void main(String args[])
    {
6       Point p = new Point();
    }
8 }

```

7. (12 points) Some aspects of each of the following can be modeled with a graph (network) structure. Describe, in each case what the nodes would represent and what the edges would represent. In general, nodes represent concepts/entities, and edges explain what the relationship is between the entities. For example, for navigation, nodes would represent locations (cities, towns, ...) and edges between cities represent if a road exists between them (a weight can be added to denote the distance).

- (a) Airline flights
- (b) Social networks such as Facebook
- (c) The Internet
- (d) Movie industry
- (e) Avengers (the movie/comic)
- (f) Amazon.com recommendation system (where a system recommends to you what to buy)

8. (10 points) Describe and specify the order of growth of each of the following code sections, using big-O notation (explain your answer!):

```

(a) int count = 0;
    2 for (int i = 1; i <= N; i++)
        count ++;

(b) int count = 0;
    for (int i = 1; i <= N; i++)
3       for (int j = 1; j <= N; j++)
            count ++;

```

```

(c) int count = 0;
    2 for (int i = 1; i <= N; i++)
        count ++;
    4 for (int j = 1; j <= N; j++)
        count ++;

(d) int count = 0;
    for (int i = 1; i <= N/2; i++)
    3     for (int i = 1; i <= N/2; i++)
        count ++;

(e) public static String reverse ( String s) {
    2 int n = s. length ();
    char [] a = new char [n];
    4 for (int i = 0; i < n; i++)
        a[i] = s. charAt (n-i -1);
    6 String reverse = new String (a);
    return reverse ;
    8 }

```

9. (5 points) Describe the order of growth of each of the following functions using O notation

- (a) $N^2 + 3 \cdot N$
- (b) $3 \cdot N + N$
- (c) $N \cdot (N - 1) + 2$

2 Programming exercises (45 points)

For the programming exercises, please follow the provided instructions in the instructions.pdf. Each of the exercises below should be a separate zip of a Java project with the name described in the instructions.

1. (15 points) Create two methods that calculate the exponentiation of a number, such as 2^3 or 4^3 . It should take the following two parameters: 1) the base of the exponent (in 2^3 the base is 2), and 2) the exponent (in expression 2^3 it corresponds to 3). Naturally, your method should work with any pair of numbers. Implement in the following two ways:

- (a) Use a for loop to exponentiate a number.
- (b) Don't use a for loop.

What is the difference between these two methods in terms of time efficiency.

2. (30 points) Create a class that models a standard pair of dice in the following way:

- (a) Create a class called *PairOfDice* in Java. Objects of this class represent a single pair of six-sided dice. The only attributes of such an object are the face values of the pair of dice. Provide a constructor. Provide a *roll* method that simulates rolling the dice (you need to use a random number generator). Provide a *value* method that returns the sum of the current face values of the dice. Provide a *toString* method that returns a nicely formatted string representing the pair of dice, for example “5 : 3 = 8”. Finally, create a “test driver” class containing the main method that demonstrates that your *PairOfDice* class performs correctly.
- (b) Now imagine that we also want to include an unfair or biased dice in our solution. This means that rolling the dice doesn’t have equal probabilities for each possible value (in a fair dice each value has the equal probability of 1/6 to be rolled). **You don’t need to implement the method of rolling an unbiased roll.** The question is how would you reorganize your program to include this *BiasedDice-Pair* class? Add that new class to your project and leave the rolling method empty if you don’t want to implement it.