King Fahd University of Petroleum and Minerals
ICS Department

ICS-202 Data Structures

Assignment 1

Spring Semester 172

Student ID	
Student Name	

ICS 202 – Data Structures
Spring Semester 2017/2018 (172)
Assignment # 1
Due on Saturday February 3, 2018 before midnight

Question I (20 points)

Design an abstract class **Student**. A student has the following information: **ID** and **GPA**. The student class has an abstract method **getStatus** that returns the status as a string and a non-abstract final method **displayStudent** that prints the details of a student. Design two subclasses **Undergraduate** and **Graduate**. The **status** of the graduate student is **good** if his GPA is 3.0 or above otherwise it is **probation**. The undergraduate's **status** is **honor** if his GPA is 3.0 or above, **good** if his GPA is 2.0 or above, **probation** otherwise. Write a test class that randomly generates 10 students and prints their information.

Program doesn't compile <max 5/20>

Compiles but doesn't run <max 8/20>

Program runs —— 10 pts

5 different test cases —— 5x2

Total : 20

-3 pts for not adhering to specs.

Question II (25 points)

Write a Fraction class that defines adding, subtracting, multiplying, and dividing fractions. Then write a method for reducing factors and methods for inputting and outputting fractions. Implement a main program that allows you the following:

a. Enter one fraction or two fractions.
b. Carry out an operation on the entered fractions (adding, subtracting, multiplying or dividing) and displays the whole operation and the result to the user. In case one fraction is entered, just display that fraction.
c. All fractions that you output must be shown in their simplest forms. For example, when entering 2/6, it is displayed as 1/3. When a fraction is entered as 0/x where x is any nonzero integer, it should be shown as 0/1.
d. Make sure you properly handle division by zero.

program doesn't compile < max 5/25 >

---

program compiles but doesn't run < max 8/25 >

---

program runs ———— 10 pts.

handles division by zero ———— 2 pts.

represents fractions $\frac{0}{a}$ as $\frac{0}{1}$ ———— 2 pts.

simplifies fraction results ———— 2 pts.

Three different test cases ———— 3x3pts

< out of 25 >

Total

Question III (55 points)

A veterinary office wants to store information regarding the kinds of animals it treats. Data includes diet, whether the animal is nocturnal, whether its bite is poisonous (as for some snakes), whether it flies, and so on. Use a superclass Pet with abstract methods and create appropriate subclasses to support about 10 different animals of your choice, with, at least, the following functionalities:

a. Ability to admit a pet into the office. Note that you need to have information of the owner stored!
b. Ability to list all pets an owner has.
c. Ability to list all pets of a certain kind (e.g. cats, poisonous, reptiles).
d. Ability to list all information regarding a certain pet.

1. (10 points) Design a class diagram using the best object-oriented practices to solve the above problem. Make sure you show each class, the instance variables and the methods defined.
2. (25 points) Implement a java program, based on your design in part 1. It is up to you whether to choose a graphical user interface or a command-based interface.
3. (10 points) Develop a suitable main program that will test your implementation.
4. (10 points) Provide a README file where you include a set of data that can be used by your main program to test it.

1. <10, 8, 5, 0>

3. <10, 9, 7, 4>

4. <10, 8, 4>

2. a. support of 10 different animals <5 points>

b. 5 different test cases to show that the program properly displays results.
5X <4, 2>

Question IV (30 points)

Answer the following questions:

1. (20 points) Given the code segment below and that ***n*** is the problem size, answer the following questions:

```
// . . .
int sum = 0;

for(int i = 1; i < n; i = i + 2)
    sum++;                                  // statement1

if(x < 5){
    for(int i = 1; i <= n; i = i * 2)
        for( int k = 1; k <= i; k = k + 1)
             sum++;                         // statement2
}
else{

     for(int i = 1; i <= n * n; i = i * 2)
           sum++;                           // statement3
}

for(int i = 1; i < n; i++)
    for( int k = 1; k <= n; k = k * 4)
        sum++;                              // statement4
```

   a. (12 points) Find the exact number of times `statement1`, `statement2`, `statement3` and `statement4` get executed.

Statement 1: $i: 1, 1+2, 1+2(2), \ldots, 1+j(2) = n-1$

$2j = n-2$

$j = \frac{n-2}{2}$

<3,2,1>

$\boxed{\text{where } i = 1+2j}$

Let $j: 0, 1, 2, \ldots, \frac{n-2}{2}$

$\therefore$ Statement 1 is executed $\sum_{j=0}^{\frac{n-2}{2}} 1 = \frac{n-2}{2} + 1$

$= \frac{n}{2}$.

Statement 2 (S2): $i: 1, 2, 2^2, \ldots, 2^j = n$

$j = \lg n$

Let $j: 0, 1, 2, \ldots, \lg n$ $\boxed{\text{where } i = 2^j}$

S2 is executed $\sum_{j=0}^{\lg n} \sum_{k=1}^{i} 1 = \sum_{j=0}^{\lg n} i$

$= \sum_{j=0}^{\lg n} 2^j = \frac{2^{\lg n+1} - 1}{2-1} = 2n - 1$

<3,2,1>

Statement 3 (S3):

<3, 2, 1>

$i = 1, 2, 2^2, 2^3, \dots, 2^j = n^2$ $\quad j = \lg n^2 = 2\lg n$

Let $j = 0, 1, 2, \dots, 2\lg n$, $\boxed{\text{where } i = 2^j}$

$\therefore$ S3 is executed $\sum_{j=0}^{2\lg n} 1 = 2\lg n + 1$.

Statement 4 (S4):

<3, 2, 1>

$k = 1, 4, 4^2, 4^3, \dots, 4^j = n$ $\quad j = \log_4 n$

Let $j = 0, 1, 2, 3, \dots, \lg_4 n$ $\boxed{\text{where } k = 4^j}$

$\therefore$ S4 is executed $\sum_{i=1}^{n-1} \sum_{j=0}^{\lg_4 n} 1$

$$= \sum_{i=1}^{n-1} (\lg_4 n + 1)$$

$$= (n-1)(\lg_4 n + 1).$$

b. (4 points) Determine the Big-O complexity of this program fragment in the best case. You must show the details of your computations and state the Big-O rules that are used in the computations.

The best case occurs when the statement with least number of operations is executed between S2 and S3. Since $cost(S2) = 2n-1$ and $cost(S3) = 2\lg n + 1$, the best case occurs when S3 is executed.

$$\therefore \text{Best case} = \frac{n}{2} + 2\lg n + 1 + (n-1)(\lg_4 n + 1)$$

<4, 3, 2>

$$= O(n) + O(\lg n) + O(1) + O(n \lg n) + O(n) - O(\lg n) - O(1) \quad [\text{using Fact 5}]$$

$$= O(n \lg n) \quad [\text{using Fact 6}]$$

c. (4 points) Determine the Big-O complexity of this program fragment in the worst case. You must show the details of your computations and state the Big-O rules that are used in the computations.

<4, 3, 2> Similar to b, and using the same rules,

$$= O(n \lg n).$$

2. (4 points) Find functions f_1 and f_2 such that both $f_1(n)$ and $f_2(n)$ are $O(g(n))$, but $f_1(n)$ is not $O(f_2(n))$.

<4, 2> $f_1(n) = n^2$ and $f_2(n) = n$. Both of them are $O(n^2)$. However, $n^2 \neq O(n)$.

3. (6 points) Determine whether the following statements are true or false. Briefly justify your answer.

a. If $f(n)$ is $\Theta(g(n))$, then $2^{f(n)}$is $\Theta(2^{g(n)})$.

False. $\lg n = \Theta(\lg n^2)$. However, $2^{\lg n} = n \neq \Theta(2^{\lg n^2})$
$= n^2$.

+1 +1

b. $f(n) + g(n)$ is $\Theta(\min(f(n), g(n))$.

False. $f(n) = n$, $g(n) = 1$.

+1 $f(n) + g(n) = n+1 \neq \Theta(1)$. +1

c. 2^{na} is $O(2^n)$, where a is a constant.

False. $2^{na} = (2^n)^a \neq O(2^n)$ since using $2^n = m$

+1 $m^a \neq O(m)$. +1