United International **University (UIU)**

Dept. of Computer Science & Engineering (CSE)

Final Exam :: Spring 2021
Course Code: CSE 1115   Course Title: Object Oriented Programming
Total Marks: **40**   Time: **1hr 45 min**

**READ THIS CAREFULLY: Any examinee found adopting unfair means will be
expelled from the trimester / program as per UIU disciplinary rules**

Question 1 (7.5) [CO2]

Read input from a text file named "in.txt" and output **only the last 5 lines** to **another file** named "out.txt". You can assume the input file contains more than five lines and each line contains a string (without any white space). **You need to open/close files properly and also handle relevant exceptions while reading from/writing to any file**.

in.txt	out.txt
line1	line3
line2	line4
line3	line5
line4	line6
line5	line7
line6	
line7	

Question 2 (2.5 + 5) [CO1]

A) The following code takes 3 integers input from the user 10 times in a for loop. The user might input int/double/String data. Rewrite the following code with appropriate try-catch blocks. Also, add a **finally** block that prints "Done!".

```java
public class ExceptionTest {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] data = new int[10];
        for(int i = 0; i < data.length; i++) {
            int a = sc.nextInt();
            int b = sc.nextInt();
            int c = sc.nextInt();
            data[a] = b / c;
        }
    }
}
```

B) Create a class **Account** that contains **two** instance variables (String **name**, double **balance**). The class also contains a method **withdraw** that takes a double parameter **amount** and reduces the **balance** value by that **amount**, if there is sufficient balance. Otherwise, the withdraw method **throws** a user defined exception called **InsufficientBalance**. InsufficientBalance class's **constructor** takes **two** parameters: the **current-balance** and **withdraw-amount**. The InsufficientBalance class sets the exception message (using super call in the constructor) as follows:

Insufficient Balance. Current balance 100 is lower than the withdraw amount 500.

Here, the constructor parameters, **current-balance** value is 100 and **withdraw-amount** value is 500**.**
Now, write the classes **Account** and **InsufficientBalance**.

Question 3 (5) [CO2] [Answer any one of the following two parts]

Part A (5)

Consider a class named "**Movie**" which has the following attributes: String **name** and double **imdbRating**. You have to write a code that contains an **Arraylist** of **5 or more** Movie objects. Then **sort** the ArrayList according to **imdbRating** in **descending** order. You can create the Movie objects in the code, then add those to the ArrayList.

Here are some example movie data:

8.6 Inception
9.1 The Shawshank Redemption

Here, the decimal number is the imdbRating and after that, the string is the movie name.

OR, Part B (2 + 3)

(i) What is the difference between wait() and sleep() methods? Give an example.

(ii) In multithreaded programs, often the main thread must be the last thread to finish. But this is not always the case. How can you ensure that the main thread definitely finishes last? Write a java code that does so. (You must use either the Runnable interface or the Thread class).

Question 4 (2 + 3 + 2.5) [CO1]

A) Point out the errors in the following code snippets (You do not have to rewrite the whole code; you can just write the lines where modifications are required). Give reasons for your answer.

Code-I	Code-II
| ```
1 class HelloWorld{
2     static int a;
3     static int b;
4     int c;
5     void firstMethod() {
6         this.a = 5;
7     }
8     static void secondMethod() {
9         System.out.println("b = " + b);
10        System.out.println("c = " + c);
11        this.a = 5;
12    }
13 }
``` | ```
1 final class A{
2     final x;
3     final void method() {
4         System.out.println("This is a final method");
5     }
6 }
7 class B extends A{
8     void newMethod() {
9         System.out.println("This is not a final method");
10    }
11 }
12 class C{
13    final void anotherMethod() {
14        System.out.println("I am inside another method");
15    }
16 }
17 class D extends C{
18    void anotherMethod() {
19        System.out.println("This is a new version " +
20                "of another method");
21    }
22 }
``` |

B) Point out the errors in the following code and write down the reason behind the errors (there are three errors). You don't have to rewrite the code:

```
class BankAccount{
    double balance;
    BankAccount(double balance){
        this.balance = balance;
    }
    void updateBalance(double newBalance){
        balance += newBalance;
    }
}

public class MainClass{
    public static void main(String[] args){
        BankAccount account = new BankAccount(){
            void updateBalance(){
                System.out.println("The balance has been updated");
            }
        };
        account.updateBalance("0");
        account.updateBalance(0);
        account.updateBalance();
    }
}
```

Question 5 (5) [CO2]

Find and fix the errors in the following code. You cannot add or remove any lines; you can only modify them. No need to rewrite the whole code, you can only provide the line number and the modified statement. After fixing the errors, add necessary codes (line 17) so that the GUI looks like the given GUI.

```java
1 public class GUIApp {
2     public GUIApp(){
3         JFrame frame = new JFrame("MyApplication");
4         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
5         frame.setDimensions(400, 150);
6         frame.setLayout(new FlowLayout());
7
8         JButton upButton = new JButton();
9         upButton.set("UP");
10
11         JLabel middleLabel = new JLabel();
12         middleLabel.set("A JLabel object can display either text, " +
13                 "an image, or both.");
14
15         JTextField downTextField = new JTextField();
16
17         // Your code here
18
19         frame.setView(true);
20     }
21
22     public static void main(String[] args) {
23         new GUIApp();
24     }
25 }
```


Question 6 (7.5) [CO1]

Define necessary classes so that the following output of the right side can be generated from the codes of the left side:
[No need to write the following codes in the answer script. Just write the solution part.]

```java
public class Final {
    public static void main(String[] args) {
        Human neurologist = new Doctor();
        Human cs1 = new CSEngineer();
        Human civil1 = new CivilEngineer();
        Engineer cs2 = new CSEngineer();
        Engineer civil2 = new CivilEngineer();

        neurologist.study();
        cs1.study();
        civil1.study();
        cs2.study();
        civil2.study();

        System.out.println("");

        neurologist.work();
        cs1.work();
        civil1.work();
        cs2.work();
        civil2.work();
    }
}
```

Output:

Studying Human Anatomy
Solving ODE/PDE
Solving ODE/PDE
Solving ODE/PDE
Solving ODE/PDE

Dissecting Human Body
Coding in Java
Designing Structures
Coding in Java
Designing Structures