United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam, Trimester: Fall 2024
Course Code: CSE-1115, Course Title: Object Oriented Programming
Total Marks: 40, Duration: 2 hours

Any examinee found adopting unfair means will be expelled from the trimester / program as per UIU disciplinary rules.

1. Write the class D, where D is a non-abstract class presented as follows: [5]

```
interface A{
    int a();
    int b();
}
interface B extends A{
    int x();
    int y();
}
```

```
abstract class C implements B{
      public abstract void p();
}
public class D extends C{
// Write your code here
}
```

The methods must implement some calculations as described below:

- a() calculates u + v.
- b() calculates u×v.
- x() calculates the remainder, where u is the dividend and v is the divisor.
- y() calculates u^v.
- p() executes all the above methods and displays the calculated results.
- Use necessary field variables and set methods. Make a main method and display all the results using u = 4, v = 10.

2. Write the InvalidAgeException class, which extends the Exception class and invokes the parent class constructor. [5]

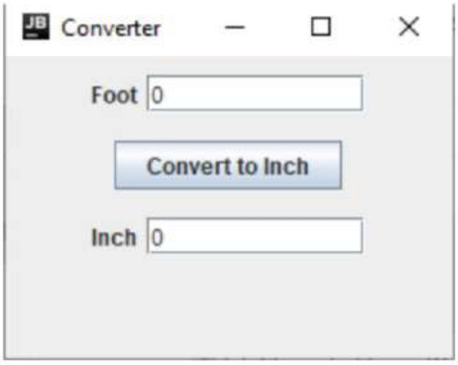
```
public class UIU {
  public static void validate(int age) throws InvalidAgeException {
       if (age < 18) {
         throw new InvalidAgeException("Age must be 18 or above");
       }
       System.out.println("Valid age: " + age);
  }

public static void main(String[] args) {
   try {
      validate(12);
   }
   catch (InvalidAgeException e) {
     System.out.println("Caught Exception: " + e.getMessage());
   }
  }
}
```

3. Imagine you have a file named "input.txt" containing various texts. Your task is to extract only the digits from this file and write them into a new file called "output.txt. [10]
Check the following example for clarification:

input.txt	output.txt
abc123ef9d	1239

To retrieve a character from a string, one can utilize the method **charAt(int index)**. For example:
String s = "hello";
char ch = s.charAt(4); // returns the char 'o'.

4. Write the Java code for the following GUI application that has the functionality of converting *feet* to *inches* (Formula: 1 foot = 12 inches). Feet will be converted to inches after pressing the *Convert* button. [10]



5. Answer question 5a or 5b (answer one question, not both):

a. Write a Java program to evaluate z given as

$$z = 1 - \frac{x}{2} + \frac{x^2}{3} - \frac{x^3}{4} + \frac{x^4}{5} \ldots + \frac{x^{50}}{51}$$

using two threads. One thread will handle the addition of negative parts, while the other will handle the addition of positive parts. The final result should be computed by combining the results from both threads, using x = 1.15.

[10]

b. Read the following **incomplete** Java program and the **expected output** carefully. Then, answer the questions below. [4 + 4 + 2 = 10]

Code

```java
import java.util.*;

class Cow implements Comparable<Cow> {
  String name;
  int price;

  public Cow(String name, int price) {
    this.name = name;
    this.price = price;
  }

  public String toString() {
    return
      name + " costs " + price + " taka.";
  }

  public int compareTo(Cow other) {
    // answer to question (i) here
  }
}

// answer to question (ii) here

class Main {
  public static void main(String[] args) {
    ArrayList<Cow> cows = new ArrayList<>();
    cows.add(new Cow("Raja Babu", 2700000));
    cows.add(new Cow("Hero Alom", 1500000));
    cows.add(new Cow("Sultan", 1800000));
    cows.add(new Cow("Minister", 1200000));
    cows.add(new Cow("Shahid", 2200000));

    System.out.println("Before Sorting:");
    for (Cow c : cows) {
      System.out.println(c);
    }

    Collections.sort(cows);

    System.out.println("\nAfter Sorting by Price:");
    for (Cow c : cows) {
      System.out.println(c);
    }

    // answer to question (iii) here

    System.out.println("\nAfter Sorting by Name:");
    for (Cow c : cows) {
      System.out.println(c);
    }
  }
}
```

Output

```
Before Sorting:
Raja Babu costs 2700000 taka.
Hero Alom costs 1500000 taka.
Sultan costs 1800000 taka.
Minister costs 1200000 taka.
Shahid costs 2200000 taka.
```

```
After Sorting by Price:
Minister costs 1200000 taka.
Hero Alom costs 1500000 taka.
Sultan costs 1800000 taka.
Shahid costs 2200000 taka.
Raja Babu costs 2700000 taka.
```

```
After Sorting by Name:
Hero Alom costs 1500000 taka.
Minister costs 1200000 taka.
Raja Babu costs 2700000 taka.
Shahid costs 2200000 taka.
Sultan costs 1800000 taka.
```

(i) Complete the `compareTo` function in the `Cow` class so that `Collections.sort(cows)` sorts the cows by price (low to high).

(ii) Write a class that implements the `Comparator` interface which can be used to sort the cows by name (alphabetical order).

(iii) Complete the `main` function to sort the cows using the comparator you wrote in (ii).

- **You do not need to write the entire code, only write the necessary part.**