



United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam, Trimester: Spring 2023

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. Answer the following questions:

- a) What is 'Abstraction' in object-oriented concept? Give an example of an Abstract class. [2]
- b) Consider 2 *interfaces* **A** and **B**. There is also an abstract class **C**. [8]
 - A has method p, q (**2 methods**)
 - B has method x, y (**2 methods**)
 - C has only one abstract method z
 - C inherits from interface A and B
 - Create a concrete class called 'D' that inherits from 'C'.

Now **write** the required code to create all the interfaces and classes. The concrete methods should print their own name. All the methods are **public** and should have a return type of 'int'.

2. Consider the following code:

```
class Math{
public int divide(int a, int b){
    return a/b;
}
}
public class test {
    public static void main(String[] args) {
        Math m = new Math();
        int n = 4;
        int d = 0;
        System.out.println(m.divide(n,d));
        System.out.println("Complete"); //this line should be always
executed
    }
}
```

- a) Write the necessary codes **within the main method** so that no exception is thrown. Also make sure "Complete" is always printed. [3]
 - b) Write a new user defined exception called '**DivbyZero**' as a subtype of '**ArithmeticException**'. [5]
 - The '**divide()**' should check if the second parameter is '**0**' or not.
 - If the second parameter is '**0**', the method should throw '**DivbyZero**' exception with a message stating "**The denominator can not be zero!**".
 - The '**divide()**' method should have proper use of '**throw**' and '**throws**'.
 - c) Briefly discuss about 'Call Stack'. [2]
3. Write a program for a large company to find the employees who are eligible for an increment. Do the following operations.
- a) Take n number of employees' name, id, and last 6 months' performance scores as user input from keyboard. [2]
 - b) Now calculate each employee's average score and write all the employee's information in a file named 'employee.txt'. Put each information in separate lines as follows- [3]
 - Employee1 name
 - Employee1 id
 - Employee1 average score
 - Employee2 name
 - Employee2 id

Employee2 average score

.....

- c) Now read the average score of each employee from the 'employee.txt' file. Find out which employee scored more than 75 and write down their names on a separate file called 'increment.txt'. [3]
- d) Find the employee/employees with the maximum score and print their names along with the score. [2]

Answer one from the question 4-6.

- 4. Write a Java program that creates a thread using **Runnable interface** to print the numbers from 1 to 10. The program should spawn the thread, start it, and then wait for it to complete. [10]
- 5. a) Suppose, you are given a string. Now create an ArrayList of characters containing each character of the string. Then change the third element(if it exists) of the ArrayList to 'z'. The program should then print out the contents of the list. [help: use set(index, element) function to set an element] [4]
b) You are given a class "Point" having two instance variables x and y. You are also given an ArrayList of type "Point". Write necessary codes to sort the ArrayList with respect to "x + y" in descending order using the "Collections.sort(<ArrayList>)" function. [6]
- 6. Consider the following code:
package Anonymous;
interface Sample {
 int x = 10;
 void sample();
}
public class Outer {
 public static void main(String[] args) {
 // code 1: create object of class Inner from Outer class
 SampleClass s=new SampleClass();
 s.sample();
 //code 2: write anonymous inner class here
 //print a and b
 }
 class Inner {
 int a = 100;
 double b = 10.0;
 }
}
class SampleClass implements Sample{
 @Override
 public void sample(){
 System.out.println("sample is="+x);
 }
}
a) i. Create an object of the class Inner from the Outer class and using the object print the variable of the class. [2+3]
 ii. Rewrite the code by using anonymous inner class.

[P.T.O]

- b) Find the output for the following code.

```
class OuterDemo {

    static int a = 100;
    String s1 = "Programming";
    String s2 = "Java";
    static double b = 15.5;

    public static void main(String[] args) {
        OuterDemo outer = new OuterDemo();
        System.out.println(outer.s2 + a + b + outer.s1);
    }

    OuterDemo() {
        Inner1 inner1 = new Inner1();
        Inner2 inner2 = new Inner2();
        System.out.println(inner1.s1);
    }

    class Inner1 {

        int a = 30;
        String s1 = "Object-Oriented";
        String s2 = "100";
        double b = 15.5;

        Inner1() {
            System.out.println(OuterDemo.this.s1 + s2);
            System.out.println(this.s1 + s2);
            System.out.println(OuterDemo.a * b);
        }
    }

    class Inner2 {

        String s1 = "Language";
        String s2 = "Python";

        Inner2() {

            System.out.println(this.s1+"-"+s2 + b);
        }
    }
}
```