



---

## End Semester Exam

IT602: Object-Oriented Programming and Data Structures

Date: 4<sup>th</sup> May 2022 | Timings: 2:00 PM – 3:30 PM

---

### Instructions:

- 6 questions | 40 total points | 90 minutes
- Questions 1 to 3: 10 marks each
- Question 4 to 5: 3 marks each
- Question 6: 4 marks
- **Attempt all questions!!**

For each programming question, you will support your code with **succinct** and **justified** comments in 1-2 sentences; any redundancy in comments will be penalized.

---

1. Design a class for a bank database. The database should support the following operations:

- deposit a certain amount into an account.
- withdraw a certain amount from an account
- get the balance (i.e., the current amount) in an account
- transfer an amount from one account to another

The amount in the transactions is a value of type `double`. The accounts are identified by instances of the class `Account` that is in the package `com.megabankcorp.records`. The database class should be placed in a package called `com.megabankcorp.system`.

*Subclass* The deposit, withdraw, and balance operations should not have any implementation, but allow subclasses to provide the implementation. (The transfer operation should use the deposit and withdraw operations to implement the transfer). It should not be possible to alter this operation in any subclass, and only classes within the package `com.megabankcorp.system` should be allowed to use this operation. The deposit and withdraw operations should be accessible in all packages. (The balance operation should only be accessible in subclasses and classes within the package `com.megabankcorp.system`).

2. Declare an interface called `Function` that has a method named evaluate that takes an int parameter and returns an int value.

Create a class called `Half` that implements the `Function` interface. The implementation of the method `evaluate()` should return the value obtained by dividing the `int` argument by 2.

In a client, create a method that takes an arbitrary array of `int` values as a parameter, and returns an array that has the same length, but the value of an element in the new array is half that of the value in the corresponding element in the array passed as the parameter. Let the implementation of this method create an instance of `Half`, and use this instance to calculate values for the array that is returned.

3. Write a program that reads text from a source using one encoding, and writes the text to a destination using another encoding. The program should have four optional arguments:

- The first argument, if present, should specify the encoding of the source. The default source encoding should be "8859\_1".
- The second argument, if present, should specify the encoding of the destination. The default destination encoding should be "UTF8".
- The third argument, if present, should specify a source file. If no argument is given, the standard input should be used.
- The fourth argument, if present, should specify a destination file. If no argument is given, the standard output should be used.

Use buffering, and read and write 512 bytes at a time to make the program efficient.

Errors should be written to the standard error stream.

4. The following program has several errors. Modify the program so that it will compile and run without errors.

// Filename: Temperature.java

```
PUBLIC CLASS temperature {  
    PUBLIC static void main(string[] args) {  
        double fahrenheit = 62.5;  
        /* Convert */  
        double celsius = f2c(fahrenheit);  
        System.out.println(fahrenheit + 'F' + " = " +  
            Celsius + 'C');  
    }  
  
    double f2c(double float fahr) {  
        RETURN (fahr - 32) * 5 / 9;  
    }  
}
```



5. The program below is supposed to calculate and print the time it takes for light to travel from the sun to the earth. It contains some logical errors. Fix the program so that it will compile and print the correct result when run.

//Filename: Sunlight.java

```
public class Sunlight {
    public static void main(String[] args) {
        // Distance from sun (150 million kilometers)
        int kmFromSun = 150000000;

        int lightSpeed = 299792458; // meters per second

        // Convert distance to meters.
        int mFromSun = kmFromSun * 1000;

        int seconds = mFromSun / lightSpeed;

        System.out.print("Light will use ");
        printTime(seconds);
        System.out.println(" to travel from the sun to
        the earth.");
    }

    public static void printTime(int sec) {
        int min = sec / 60;
        sec = sec - (min * 60);
        System.out.print(min + " minute(s) and " + sec +
        " second(s)");
    }
}
```

6. A palindrome is a text phrase that is spelled the same backward and forward. The word redivider is a palindrome since the word would be spelled the same even if the character sequence were reversed. Write a program that takes a string as an argument and reports whether the string is a palindrome. Use customized exceptions to handle any invalid input (e.g. a numeric value). Also, mention the time complexity of your method in terms of the length of the string.

\*\*\* End of the Paper \*\*\*