



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CSE1007 Java PROGRAMMING LAB
Winter Semester 2020-21
Challenging Task - I

DATE: 31/03/21 SLOT: L43+L44 CLASS NBR: 4170 TIME: 2:10-3:20 PM

Instructions:

After completing your program, **upload** the same in **Challenging Task-I Lab component under V-TOP login**.

Upload File should contains:

- Source code
- Sample input and output
- Output screen shot

1. a. Caesar's Code is one of the simplest encryption techniques. Each letter in the plaintext is replaced by a letter some fixed number of position (n) down the alphabet cyclically. In this exercise, we shall pick $n=3$. That is, 'A' is replaced by 'D', 'B' by 'E', 'C' by 'F', ..., 'X' by 'A', ..., 'Z' by 'C'.

Write a program called **CaesarCode** to cipher the Caesar's code. The program shall prompt user for a plaintext string consisting of mix-case letters only; compute the ciphertext; and print the ciphertext in uppercase. For example,

Enter a plaintext string: **Testing**

The ciphertext string is: WHVWLQJ

b. Design a class Student derived from a class Person. The Person class has name and age data and display method to display the name and age of a person. The Student class has data like rollNo and branch and display method to display name, age, rollNo and branch of the student. Each class have their own constructor to initialize the value of each data field. Finally create MainDemoClass and create array of objects for student class. Print the values of all objects in the MainDemoClass.

2. a. Write a program called **DecipherCaesarCode** to decipher the Caesar's code described in the previous Question 1.a. The program shall prompts user for a ciphertext string consisting of mix-case letters only; compute the plaintext; and print the plaintext in uppercase. For example,

Enter a ciphertext string: **wHVwLQJ**

The plaintext string is: TESTING

b. Write a Java program 'WordCount' that counts the words in two files. Start a new thread for each file. Assume file name as data1.txt and data2.txt, then the program might print

data1.txt: 100

data2.java: 130

3. a. This simple cipher exchanges 'A' and 'Z', 'B' and 'Y', 'C' and 'X', and so on.

Write a program called **ExchangeCipher** that prompts user for a plaintext string consisting of mix-case letters only. Your program shall compute the ciphertext; and print the ciphertext in uppercase. For examples,

Enter a plaintext string: **abcXYZ**

The ciphertext string is: ZYXCBA

b. Design a class Person that has name and age as data and display method to display the name and age of a person. Write the person object to a file and display the person with name starts with 's' from the file.

4. a. Write a program called **Hex2Bin** that prompts user for a hexadecimal string and print its equivalent binary string. The output shall look like:

Enter a Hexadecimal string: **1abc**

The equivalent binary for hexadecimal "1abc" is: 0001 1010 1011 1100

b. Design a class bank that has accname and branch as data and display method to display the accname and branch of an account. Write the account object to a file and display the branch with name 'vellore' from the file.

5. a. Write a program called **Dec2Hex** that prompts user for a positive decimal number, read as int, and print its equivalent hexadecimal string. The output shall look like:

Enter a decimal number: **1234**

The equivalent hexadecimal number is 4D2

b. Declare the JavaProgrammer class to describe a java programmer. Declare the JavaDeveloper class to describe a java developer. The JavaDeveloper should extend the JavaProgrammer class. Declare the JavaArchitect class as a class that describes a java architect and extends the JavaDeveloper. Think about the different variables and methods (constructors as well) that should be declared in each one of the classes. Test the classes that you declared using a stand-alone application (another separate class) by creating array

of objects and check whether the Programmer expert in “Python”. If so, display the programmers name.

6. a. Write a method called `exponent(int base, int exp)` that returns an int value of base raises to the power of exp. The signature of the method is:

```
public static int exponent(int base, int exp);
```

Assume that exp is a non-negative integer and base is an integer. Do not use any Math library functions.

Also write the `main()` method that prompts user for the base and exp; and prints the result. For example,

Enter the base: **3**

Enter the exponent: **4**

3 raises to the power of 4 is: 81

b. Declare the **PizzaDeluxe**, **PizzaSpecial** and **PizzaWoogy** classes. Each of these classes should extend the **Pizza** class. The **Pizza** class should implement the Comparable interface. The **Pizza** class should have the following private attributes:

- calories:double
- name:String
- price:double

The **Pizza** class should implements the Comparable interface and it should have the following public methods:

- + Pizza(double, String, double)
- + getCalories():double
- + getName():String
- + getPrice():double
- + toString():java.util.String
- + compareTo(Object):int

In the classes that extend the Pizza class you should declare - in each one of them - at least three attributes, three methods and two constructors. Write a stand-alone application that instantiates the 3 concrete classes and creates an array (the array type is Pizza) of 10 different objects from the PizzaDeluxe, PizzaSpecial and PizzaWoogy classes. The stand-alone application should sort the 10 different objects according to their Price.

7. a. Write a boolean method called `isOdd()` in a class called **OddEvenTest**, which takes an int as input and returns true if the it is odd. The signature of the method is as follows:

```
public static boolean isOdd(int number);
```

Also write the main() method that prompts user for a number, and prints "ODD" or "EVEN". You should test for negative input. For examples,

Enter a number: **9**
9 is an odd number

Enter a number: **8**
8 is an even number

Enter a number: **-5**
-5 is an odd number

b. Declare the TV class to describe a television. Declare the Monitor class to describe a monitor. The TV class should extend the Monitor class. Think about the different variables and methods (constructors as well) that should be declared in each one of the classes. Test the classes that you declared using a stand-alone application (another separate class) by creating array of objects and display Monitor inches in ascending order.

8. a. Write a boolean method called hasEight(), which takes an int as input and returns true if the number contains the digit 8 (e.g., 18, 168, 1288). The signature of the method is as follows:

```
public static boolean hasEight(int number);
```

Write a program called **MagicSum**, which prompts user for integers (or -1 to end), and produce the sum of numbers containing the digit 8. Your program should use the above methods. A sample output of the program is as follows:

Enter a positive integer (or -1 to end): 1
Enter a positive integer (or -1 to end): 2
Enter a positive integer (or -1 to end): 3
Enter a positive integer (or -1 to end): 8
Enter a positive integer (or -1 to end): 88
Enter a positive integer (or -1 to end): -1
The magic sum is: 96

b. Declare the School class to describe a school. Declare the University class to describe a university. The University class should extend the School class. Think about the different variables and methods (constructors as well) that should be declared in each one of the classes. Test the classes that you declared using a stand-alone application (another separate class) by creating array of objects and check whether university name "VIT" exist.

9. a. Write a boolean method called **copyOf()**, which takes an int Array and returns a copy of the given array. The method's signature is as follows:

```
public static int[] copyOf(int[] array)
```

Also write a test driver to test this method.

Write another version for **copyOf()** which takes a second parameter to specify the length of the new array. You should truncate or pad with zero so that the new array has the required length.

```
public static int[] copyOf(int[] array, int newLength)
```

b. Declare the `pgmspecialization` class to describe a programme specialization (as BCE, BCB) in our university. Declare the `PrmGrp` class to describe Programme Group (as BTech, Mtech). The `pgmspecialization` class should extend `PrmGrp` class. Think about the different variables and methods (constructors as well) that should be declared in each one of the classes. Test the classes that you declared using a stand-alone application (another separate class) by creating array of objects and display number of specialization available in a programme group.

10. a. Write a method called **reverse()**, which takes an array of int and reverse its contents. For example, the reverse of [1,2,3,4] is [4,3,2,1]. The method's signature is as follows:

```
public static void reverse(int[] array)
```

Also write a test driver to test this method.

b. Write a java application class to accept 25 student marks from the standard input device (acceptable range of marks is 0 to 100) and store them in the file "mark.txt". Include a block in the same main method to read the contents of the file "mark.txt" and display the count of students who have secured less than 50 marks. Store the marks less than 50 in the file "mark1.txt" and others in "mark2.txt". Use appropriate stream classes and file names must be passed as command line arguments to the class. Finally print the average marks those who have scored >50.