JA111 C3 Evaluation

Q1) Write a program that takes input from the user that is size of the array and the elements in the array.
The program then asks the user to enter a particular index and prints the element at that index. Index starts from zero.
This program may generate Array Index Out Of Bounds Exception or NumberFormatException. Use proper try catch mechanisms to handle this exception.

Sample Input 1:

```
Enter the number of elements in the array
2
Enter the elements in the array
50
80
Enter the index of the array element you want to access
1
```

Output 1:

```
The array element at index 1 = 80
The array element successfully accessed
```

Sample Input 2:

```
Enter the number of elements in the array
2
Enter the elements in the array
50
80
Enter the index of the array element you want to access
9
```

Output 2:

```
java.lang.ArrayIndexOutOfBoundsException
```

Sample Input 3:

```
Enter the number of elements in the array
2
Enter the elements in the array
50
80
Enter the index of the array element you want to access
F
```

Output 3

```
java.lang.NumberFormatException
```

Q2)Create an Employee class with the following protected fields:

employeeId: Integer
employeeName: String
salary: double

Use appropriate getter setter methods.

Write a public 2 argument constructor with arguments – employeeId, and employeeName.

Write an abstract method inside the Employee class
calculateSalary: void

Create another class PermanentEmployee as a child of the above Employee class and inside this class define one private field

basicPay: double

Define a 3 parameter constructor inside this class to take (employeeId, employeeName and basicPay).

Implement the calculateSalary method in PermanentEmployee class as
salary = basicPay – PF amount;
Set this value to the salary attribute.

Here PF Amount = basicPay * 0.12

Create another class TemporaryEmployee as a child of Employee class with the following private fields:

hoursWorked: Integer
hourlyWages: Integer

Define a 4 argument constructor with arguments – employeeId, employeeName, hoursWorked and hourlyWages.

Implement the calculateSalary method in TemporaryEmployee class as
salary = hoursWorked * hourlyWages
Set this value to the salary attribute.

Develop a class Loan inside this Loan class define a method calculateLoanAmount as follows:
public double calculateLoanAmount(Employee employeeObj)

This method should calculate the loan amount and return that amount.
Provide the implementation for this method as mentioned below

Loan amount is calculated as follows :If the Employee object is of type PermanentEmployee then loan amount should be
15% of the salary.

If the Employee object is of type TemporaryEmployee then loan amount should be
10% of the salary.

Note: Inside the Loan class make sure to have a private constructor

Define a Main class with the main method and inside the main method, get the Loan class object and call the calculateLoanAmount() method 3 times:

1. by supplying PermanentEmployee object
2. by supplying TemporaryEmployee object
3. by supplying a null value

and display the appropriate result.

Q3)Create an enum called Month with the all month from JAN to DEC.

Create a class Demo that define a method called :

showDetails(Month m): void

inside this method switch on the Month enum and display the details of the Month :

Example:
If Month = JAN then print “This is the 1st Month of the Year January”
If Month = FEB then print “This is the 2nd Month of the Year February”
as it is till Month DEC

Inside the main method, call the above showDetails() method by taking the input month name from the user.
if the user supplies any wrong month name then handle the exception and print the proper message : Invalid Month Name

Q4)Create a Customer bean class with the following fields:

username: String
password: String
mobileNumber: String
email: String

Create a Demo class with the main method and perform the following tasks:

Take the input from the user and create an object of the Customer with all the details.

Make sure that to perform the following input validations:

username: should contain only alphabets and length should be min 3 and max 8 characters.
password: should be alpha-numeric and also min 3 and max 8 characters.
mobileNumber: should be an Indian mobile number starting with 6,7,8,9 and the length should be 10.
email: a valid email address.

If all input fields are valid then only create an object of Customer and print all the details from the Customer object.
otherwise, print the appropriate validation error message.