Espire_InductionTraining_JavaTrack_August_2022			
Weekly Assignment Week 3			
Prepared by – Prasad Deshmukh			
Sr.	Coverage	Task	Points
1	**Java Collection, Date API**	Arjuna is developing Bill payment system for his payment agency. He wants the application to auto deduct the bills from Customer's account and present the summary of account and bill along with customer details. Lets see how he is developing this application. Develop a software application with following specifications: a. Create public class Account with private instance variables: acctno(int) and balance(double). b. Add default and parameterized constructors and do necessary initialization. c. Add getters and setters for all fields. d. Add overridden toString() method to return string format of Account object. e. The string format should be: "AccountNo: %-10s\tBalance:$%-10s" f. Create public class Customer with private instance variables : id (int), name (String), phone (String) and account (Account). g. Add default and parameterized constructors to do necessary initialization. h. Add setters/getters for all fields. i. Add toString() method to return String format of Employee object using format: "Customer details:\nCustomer Id: %-10s\tName: %-10s\tPhone: %-10s\nAccount details:\n%-10s". j. Create public class Bill with private instance variables: id(int), custid(int), billGenerationDate(Date), billPaymentDate(Date), amount(double) and paid(boolean). k. Add default and parameterized constructors and do necessary initialization. l. Add getters and setters for all fields. m. Add overridden toString() method to return string format of Account object.	25

n. The string format should be:"Bill details:\nBill Id: %-10s\tCustomer Id: %-10s\tAmount:$%-10s\tGeneration Date: %-10s\tPayment Date: %-10s\tPaid: %-10s\n"

o. Use appropriate date format to show the dates.

p. Create public class Transaction with members custList(List<Customer>), billList(List<Bill>), bill (Bill) and customer(Customer).

q. Add default and parameterized (2 parameters: List<Customer> and List<Bill> types) constructors to do necessary initialization

r. Add following methods:

Method name	Return type	Parameters	Description
getCustomer()	Customer	int id	Returns Customer instance whose id is passed as parameter to method, if not found return null.
payBill()	void		Implementation of this method shows whether bill deducted or not and will display customer, account and bill details. This method uses getCustomer() method to see whether a customer exists or not which has been mentioned in bill.

s. Create public class Main with public static void main() method to instantiate Transaction class by

passing required data and invoke its payBill() method.

Sample IO

Bill cannot be paid for customer id 1

Customer details:

Customer Id: 1   Name: Tom   Phone: 9090901010

Account details:

AccountNo: 1   Balance:$1000.0

Bill details:

Bill Id: 1   Customer Id: 1   Amount:$1000.0   Generation
Date: 01-01-2022 Payment   Date: Paid: Not paid

Bill paid for customer id 2

Customer details:

Customer Id: 2   Name: Jerry   Phone: 9090902020

Account details:

AccountNo: 2   Balance:$1500.0

Bill details:

Bill Id: 1   Customer Id: 2   Amount:$500.0   Generation
Date: 10-01-2022 Payment   Date:
06-05-2020 Paid: Paid

		Customer with customer id 3 does not exists	