UNIVERSITY OF TECHNOLOGY, JAMAICA
SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY
OBJECT ORIENTED PROGRAMMING (CIT2004)

Programming Group Project

Lecturers: Mr. Panther, Mr. Clarke, Mr. Charles

Group Size: 3-4 persons per group

Given: Week of September 27, 2021. **Due**: November 12, 2021

This OOP group project is designed to allow students to employ key OOP concepts in the analysis, design, and implementation of a real-life application. Students should make use of **inheritance**, **polymorphism**, **dynamic-binding**, **encapsulation**, **data abstraction**, **persistence** through file handling, and **defensive programming using exception handling** techniques. Students in each group, **MUST** be in the same **lab** class.

Scenario Description

Currently, Jamaica has several telecommunication services providers. They have agreed to build out a shared platform that would allow for both of their customer base to interact with their shared services. All service providers have a company id, address, number of customers (static type) and allow service providers to conduct the following tasks: add customer, view customer base, create phone credit and view all phone credit. Digicel and Flow are two service providers. Digicel maintains information on the number of branches and Flow stores its parent company name. The customers for both service providers have a customer id (TRN), last name, address, telephone number and credit balance. A telephone number is made up of an area code, prefix and serial number. The program should have two main sections – an Administration section and a Customer section.

Administration

The administration section allows for the two service providers to manage their customer base and phone credit. When this section is selected, the user is given the

option to select a provider and enter a password. The password for Digicel is "TheBiggerBetterNetwork2021" and the password for Flow is "TheWayIFlow2021". The Customer information is stored in separate files for each service provider (Digicel_Customers, Flow_Customers). The card topup information for both companies are also stored in separate files (Digicel_CardInfomation, Flow_CardInformation).

Add Customer: Every time a new customer is create, their balance is initialized to $100 and that record is written to the appropriate service provider's customer file.

View Customer Base: Displays all the customers for the selected service provider.

Phone Credit Creation: Each service provide is able to create new phone card record. Each credit card has a card number (a unique 13-digit card number), denominations ($100, $200, $500 or $1000) and Status (used/available).

View Phone Credit: Displays all the phone credit details for the selected service provider

Total number of customers: Displays the total number of customers across all service providers. This field not stored to any file.

Customer

Add Credit: To add the call credit, the user dials asterisk, one, two, one, asterisk, followed by the 13-digit card number followed by another asterisk then the customers number, then lastly the number sign: **121*[card number]*[customer telephone number]#* e.g. ****121*123456789012*8766031234#***. All other formats are invalid. When credit is added, the customer's account gets updated with the appropriate amount. The phone card record is also updated to show that the card is "Used".

All Digicel customer telephone numbers begin with the prefixes: 301, 302, 303 or 304. All Flow customer telephone numbers begin with the prefixes: 601, 602, 603, or 604. All other numbers should be treated as invalid.

Check Balance: The customer is able to check their balance by dialing asterisk, one, two, zero, asterisk, followed by the customers number then lastly the number sign:

120*[customer telephone number]#* e.g., ****120*8766031234#*. The system will parse the string entered, to determine which service provider's phone credit file to check, based on the customer's telephone prefix.

Hint: research the substring( ) method.

The system should prompt the user with the following menu:

a) Administration
    i. Digicel
        a) Display company information
        b) Add customer
        c) View customer base
        d) Create phone credit
        e) View all phone credit
    ii. Flow
        a) Display company information
        b) Add customer
        c) View customer base
        d) Create phone credit
        e) View all phone credit
    iii. Display total number of customers
b) Customer
    i. Add Credit
    ii. Check Balance

Requirements

Conduct an object-oriented analysis (noun-verb analysis or textual parse) on the above description of the file types processed by the system and determine what are the appropriate classes along with their respective attributes and methods. Utilize polymorphism to make sure your code is generic and concise. Use exception handling to make your code robust – it should not crash if invalid data is entered.

Submission

Confirm with your lab tutor how submissions should be made.

Late Penalty

As per the university's policy, late submission of projects will have a penalty applied as follows:

Time of Submission after Deadline	Penalty (%)
One day	10
Two days	20
Three days	50
Beyond three days	100

Grading Scheme Break down

No.	Section/Objective	Max. Mark(s)	Act. Mark(s)
1.	**DOCUMENTATION**		
1.1	*Group Report (outlining contribution(s) of each member)*	2	
1.2	*Object-Oriented Analysis and Design (OOA&D)*	6	
1.3	*Declaration of Authorship Form for each group member*	2	
	DOCUMENTATION SECTION TOTAL	**10**	
2.	**SOURCE CODE**		
2.1	*Comments*		
2.1.1	Each file should have all programmer's id and name	1	
2.1.2	Proper use of inline and method comments where necessary	1	
2.2	*Naming Convention*		
2.2.3	Appropriate and consistent naming convention	3	
2.3	*Object Oriented Techniques*		
2.3.1	Sub-Type Polymorphism	10	
2.3.2	Composition	5	
2.3.3	Overloading	5	
2.4	*File Management techniques*	15	
	SOURCE CODE SECTION TOTAL	**40**	
3	**FUNCTIONALITY**		
3.1	*Robustness*		
3.1.1	User Input validation	2	
3.1.2	Exception Handling	3	
3.2	*System Feature(s) Implemented*		
3.2.1	Administrator Menu – Manage Customer, Cards, Total Number of Customers	20	
3.2.2	Customer Menu – Add Credit, View Balance	15	
	FUNCTIONALITY SECTION TOTAL	**40**	

4	**USER INTERFACE**		
4.1	*Ease of User Interaction*		
4.1.1	Consistent Screen Layout & Design	2	
4.1.2	Program Navigation (i.e., Menu System)	3	
4.1.3	Graphical User Interface	5	
	USER INTERFACE SECTION TOTAL	**10**	
	LATE SUBMISSION PENALTY	**0**	**-**
	FINAL PROJECT MARK	**100**	