UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

Semester II 2023/2024

Subject : SECJ1023 Programming Technique II

Section : 08

Task : Group Project Deliverable 2 Problem Analysis and Design

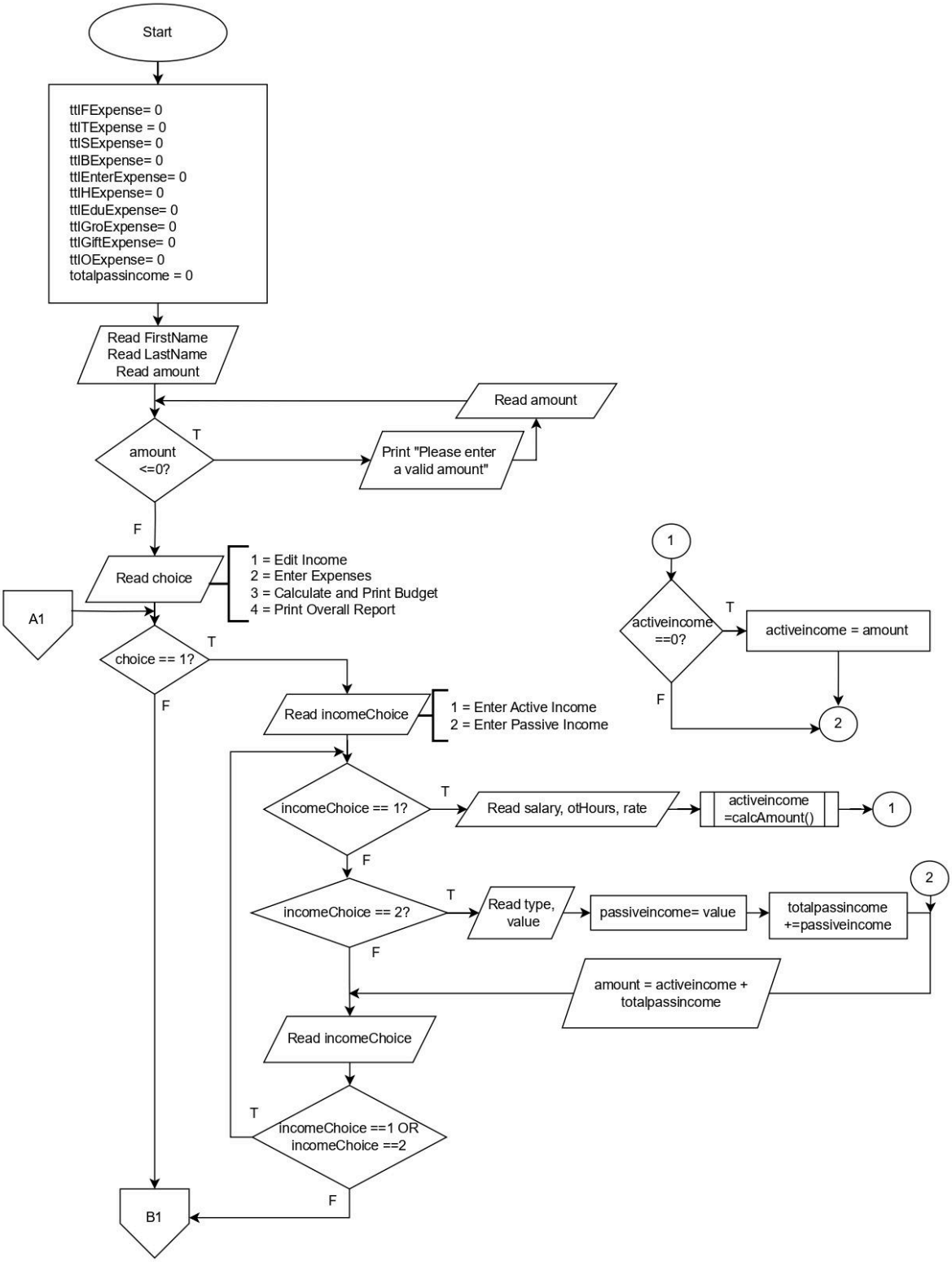
Lecturer: Dr. Lizawati binti Mi Yusuf

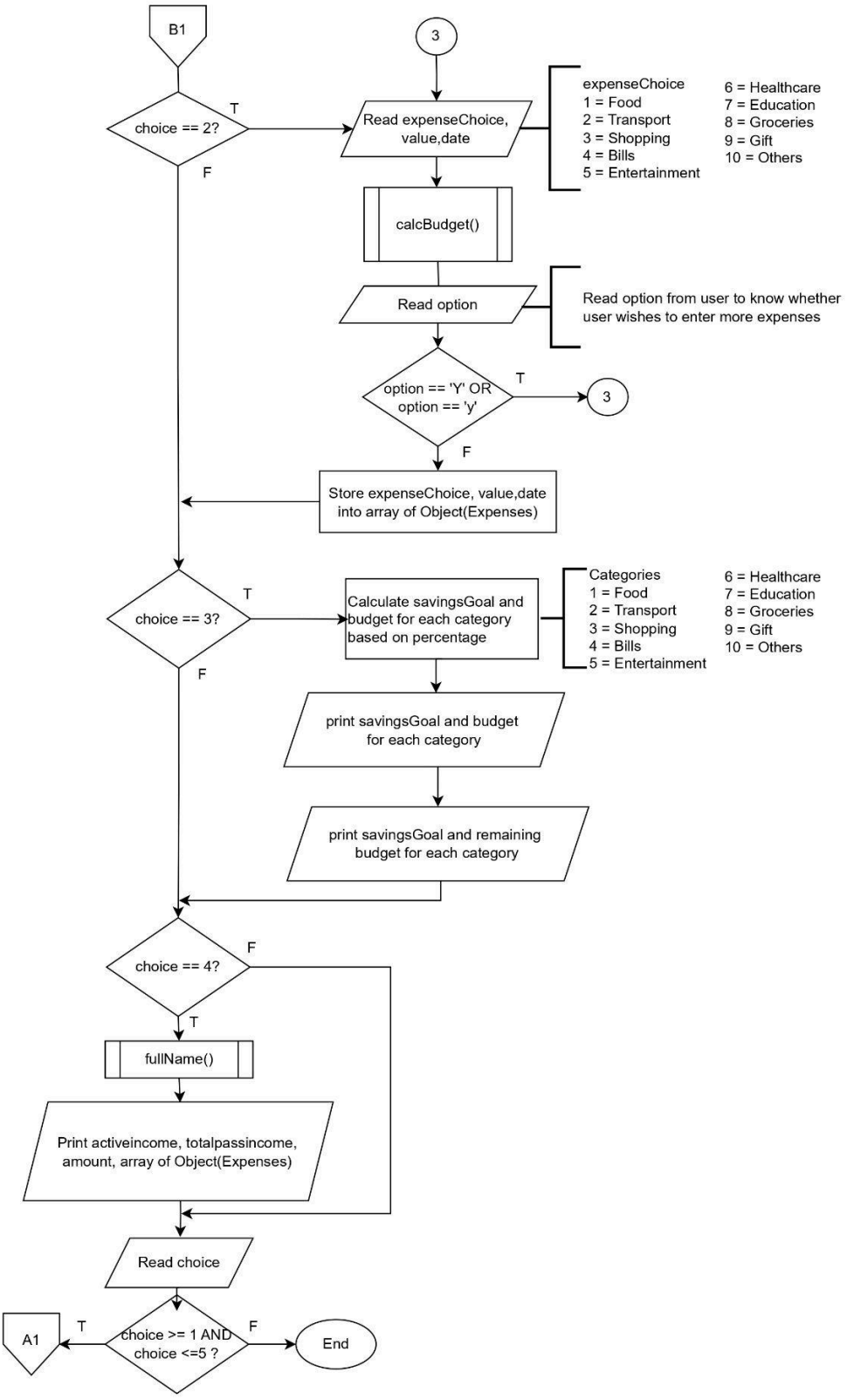
Group 7 : 3Q

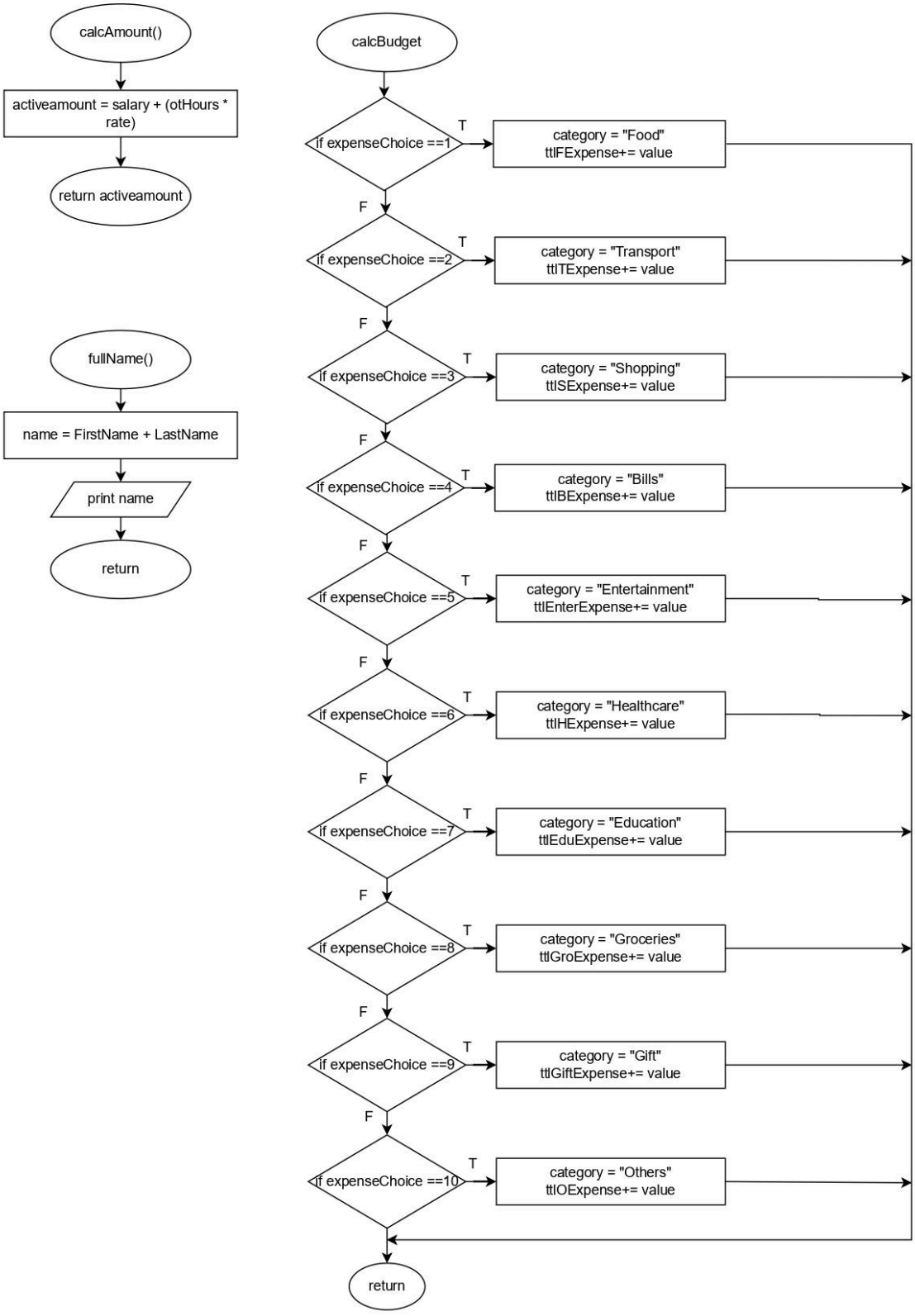
Student Name	Matric Number
Tan Keqin	A23CS0184
Mavis Lim Hui Qing	A23CS0110
Chong Lun Quan	A23CS0067

Section A: Flow Chart

I. Flow Chart




B1
choice == 2?
T
F
3
Read expenseChoice, value,date
expenseChoice
1 = Food
2 = Transport
3 = Shopping
4 = Bills
5 = Entertainment
6 = Healthcare
7 = Education
8 = Groceries
9 = Gift
10 = Others
calcBudget()
Read option
Read option from user to know whether user wishes to enter more expenses
option == 'Y' OR option == 'y'
T
3
F
Store expenseChoice, value,date into array of Object(Expenses)
choice == 3?
T
F
Calculate savingsGoal and budget for each category based on percentage
Categories
1 = Food
2 = Transport
3 = Shopping
4 = Bills
5 = Entertainment
6 = Healthcare
7 = Education
8 = Groceries
9 = Gift
10 = Others
print savingsGoal and budget for each category
print savingsGoal and remaining budget for each category
choice == 4?
F
T
fullName()
Print activeincome, totalpassincome, amount, array of Object(Expenses)
Read choice
A1
T
choice >= 1 AND choice <=5 ?
F
End


calcAmount()
activeamount = salary + (otHours * rate)
return activeamount
fullName()
name = FirstName + LastName
print name
return
calcBudget
if expenseChoice ==1
T
category = "Food"
ttlFExpense+= value
F
if expenseChoice ==2
T
category = "Transport"
ttlTExpense+= value
F
if expenseChoice ==3
T
category = "Shopping"
ttlSExpense+= value
F
if expenseChoice ==4
T
category = "Bills"
ttlBExpense+= value
F
if expenseChoice ==5
T
category = "Entertainment"
ttlEnterExpense+= value
F
if expenseChoice ==6
T
category = "Healthcare"
ttlHExpense+= value
F
if expenseChoice ==7
T
category = "Education"
ttlEduExpense+= value
F
if expenseChoice ==8
T
category = "Groceries"
ttlGroExpense+= value
F
if expenseChoice ==9
T
category = "Gift"
ttlGiftExpense+= value
F
if expenseChoice ==10
T
category = "Others"
ttlOExpense+= value
return

II. Description

Users are required to enter their name and amount(salary). If an amount is less than or equal to 0, users are required to enter the amount again until a valid amount(more than 0) is entered. Users are required to enter the choice(menu choice) where (1- Edit Income, 2-Enter Expenses, 3-Calculate and Print Budget, 4-Print Overall Report).

If choice 1 (Edit Income) is selected, users are required to enter the incomeChoice where (1-Active Income, 2-Passive Income). If incomeChoice 1 (Active Income) is selected, users need to enter salary, otHours, and rate(hourly rate). Then, activeincome will be calculated. If the activeincome is 0, we will use the amount that was entered by the user at first. If incomeChoice 2 (Passive Income) is selected, users need to enter type (type of passive income) and value (value of passive income). Then totalpassincome (total passive income) will be calculated and keep updating if users enter the passiveincome more than once. The amount will be updated by adding activeincome and totalpassincome. Users can choose to either edit the income again or choose the next choice (menu choice). If choice 2 (Enter Expenses) is selected, users are required to enter the expenseChoice (category of expenses as listed in the flow chart), value(expenses), and date(date of transaction). Then, the program will calculate the totalexpenses by adding the value entered by the user and keep updating the totallexpenses if the users choose the same expenseChoice more than once. Users can choose to either enter the value again or choose the next choice (menu choice). If users enter neither ‘Y’ nor ‘y’, the system will store expenseChoice, value, and date into an array of Objects(Expenses). If choice 3 (Calculate and Print Budget) is selected, the program will allocate the budget of each category based on the amount(salary). If choice 4 (Print Overall Report) is selected, the program will display the report to users. The report contains fullname, activeincome, totalpassincome, amount, and array of Object(Expenses). If a choice other than 1,2,3 and 4 is entered, the program will stop executing. If users enter choice 1,2,3 or 4 then the program will keep executing until the user enters other numbers.

Section B: Problem Analysis

I. Objects and Classes

Objects:

- User user
- Income income
- ActiveIncome activeIncome
- PassiveIncome passiveIncome
- Budget budget
- Expense expense
- Name name

Classes:

- User
  Attributes:
  -name: Name
  -income : Income*
  -expenses : Expense*
  -expenseCount : int
  Methods:
  -User(fname : string, lname : string) : name(fname, lname)
  -getIncome() : Income*
  -getExpenseCount() : int
  -display(user : User) : void
  -getExpense() : Expense*

- Name
  Attributes:
  -fname: string
  -lname : string
  Methods:
  -getfname() : string
  -getlname() : string
  -getFullName() : string

- Expense
  Attributes:
  -spend : double
  -date : string

–type : string
Methods:
-Expense(s : double, d : string, t : string)
-getSpend() : double
-getDate() : string
-getType() : string

- Income
  Attributes:
  -amount : double
  -source : string
  Methods:
  -Income(a : double , s : string)
  -getAmount() : double
  -getSource() : string
  -virtual display() : void

- ActiveIncome
  Attributes:
  -salary : double
  -othour : int
  -rate : double
  Methods:
  -ActiveIncome(s : double, o : int, r : double, a : double, src : string)
  -getSalary() : double
  -getHour() : int
  -getRate() : double
  -calcAmount() : void
  -display() : void

- Passiveincome
  Attributes:
  -type : string
  Methods:
  -PassiveIncome(a : double, src : string, type : string)
  -getType() : int
  -display() : void

- Budget
  Attributes:

-savingGoal: double
-myIncome : Income*
Methods:
-Budget(inc : Income*)
-setIncome(inc : Income*) : void
-calcBudget(): void
-printBudget() : void

II. Class relationships

- Association
  - Aggregation : 1) User has an Income
    - 'User' and 'Income' are independent where a 'User' may not have 'Income'. The 'User' class uses 'Income' to get the user's income details. The 'Income' object can be shared or changed without affecting the 'User' class.

    2) Budget has an Income
    - 'Budget' and 'Income' are independent where the budget may not have 'Income'. The 'Budget' class uses the 'Income' class to calculate the budget. If the 'Budget' is destroyed, 'Income' remains unaffected. A 'User' that has an income does not necessarily have a budget.

    3) User has an Expense
    - 'User' and 'Expense' are independent where a user may not have expenses.
  - Composition : 1) User consists of Name
    - 'User' contains 'Name'. This means that 'Name' objects are part of the 'User', and if the 'User' object is destroyed, its 'Name' objects are also destroyed. A 'User' must have a 'Name', else the user is not existing.

- Inheritance
  - Parent Class : Income
    We chose Income as the parent class because it contains the attributes amount and source where they are common to all types of income.
  - Child Classes :
    - ActiveIncome
      ActiveIncome is a specific type of income that comes with additional attributes and methods that are unique. It inherits the Income to gain the

common properties but also adds its specific attributes and methods. We update the amount in the Parent Class using the attributes inside the derived class(ActiveIncome) such as salary, otHours, and rate.

- Passiveincome
  PassiveIncome is a specific type of income that comes with additional attributes and methods that are unique. It inherits the Income to gain the common properties but also adds its own specific attributes and methods. We can display the amount in the Parent Class (that is updated in the User class's constructor using a for loop to calculate the total passive income).

Section C: Class Diagram