Assignment 1 – Learning the basics of Swift

Exercise 1 (10%)

You took out a loan of $1000. The details of your loan are as follows:
- Interest on your loan is compounded at a rate of 2% per month while the outstanding balance of the loan is more than $500, and at 1% per month otherwise.
- You pay back $100 every month, except for the last payment (when the balance will be less than or equal to $100)
- You pay at the end of the month, *after* the interest on the balance has been compounded.

Write a program that displays a monthly statement of the balance (after the interest is added and the monthly payment has been made). YOU DO NOT NEED THE USER TO ENTER THE AMOUNT (keep it fixed at $1,000). Your program should also display the amount of the final payment, and how many months it took to pay back the loan.
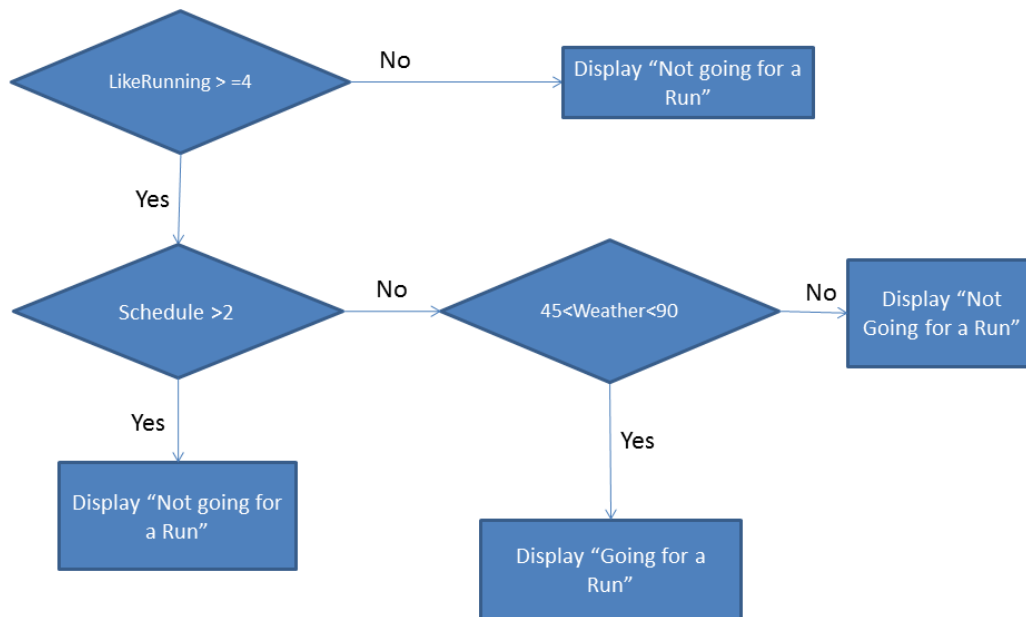
Exercise 2 (10%)

Assume the following variables describe a set of variables that have already been assigned values and are independent:
- Weather – A numeric value between 20 and 120
- Schedule – A numeric value between 0 and 5 that indicates your "business". 0 means nothing going on, 5 means "packed schedule"
- LikeRunning – A numeric value between 0 (hate running) and 10 (love running)

Write a program that will display "Going for a Run" or "Not going for a Run" based on the values of these variables.

**** Playground's latest version doesn't allow User Input. Thus, please predefine the following variables with a number: Weather, Schedule, LikeRunning ****

The logic that determines which output to display is given by the following flowchart:

Exercise 3 (10%)

A group of Stevens friends decide to run the Boston Marathon. Their names and times (in minutes) are below:

Name Time (minutes)

Elena 341, Thomas 273, Hamilton 278, Suzie 329, Phil 445, Matt 402, Alex 388, Emma 275, John 243, James 334, Jane 412, Emily 393, Daniel 299, Neda 343, Aaron 317, Kate 265

Find the fastest runner. Print the name and his/her time (in minutes).

Exercise 4 (10%)

ATM class represent the real **ATM** machines. Each **User** (actor) start a **Session** in an **ATM** machine. This relation is **an association class ATMLogin**. Create the classes in Swift.

The **ATMLogin** has the user name and password. After enter the user name a **Session** is created. With a **Session** the **User** can perform different **Transactions**.

Each **Card** belongs to an **Account** of a **Bank**. The **Bank** perform the different **Transactions**. To **simulate** the **Card** insertion a list of **Cards** is displayed to **select one**.
The **ATM** machine has **Log** of operations. The **ATM** bills are managed by the **InnerCash** handler.

*****Please create just the classes and the members. Provide only the skeleton of the methods, not the contain. *****

InnerCash
- bill1: int
- bill10: int
- bill100: int
- bill20: int
- bill5: int
- bill50: int

Card
- expirationDate: date
- holderName: char
- holderSurname: char
- number: char
- password: char
+ bank() : aBank

Account
- initialBalance: float
- number: char
- overdraftLimit: int
- owner: char

Bank
- code: char
- name: char

ATMLogin
- password: char
- user: char

ATM
- name: char
- number: char

Session
- lastMessage: char
- number: char
+ account() : anAccount
+ bank() : aBank

Transaction
- amount: float
- currentBalance: int
- date: date
- done: boolean
- number: char
- time: time
+ accont() : anAccount
+ bank() : aBank
+ card() : aCard

Log
- cardNumber: char
- date: date
- details: char
- sessionNumber: char
- time: time
- transactionNumber: char

Money
- value: float

Withdrawal

Inquiry

Deposit

Transfer
- targetAccountNumber: char
- targetBankCode: char

Currency
- country: char
- name: char
- symbol: char