**Object Oriented Programming-1**

**Bank V-4**

**You may need several extra fields for your solution**

Design a static class called "**IDGENARATOR**" with the following functionalities:

-static intserial\_no

+[p]String generate//this will automatically generates an ID for an account

the format will be "YYYY-MM-SSSSS".

example: if the serial number is 1 and date is september,2012 the auto gen ID will be "2012-09-00001".

if the serial number is 102 and date is januray,2013 the auto gen ID will be "2012-09-00102".

\*\*\* please reset the serial no to 1 if the month is changed

\*\*\* assume other variable if needed (you must need another one)

Design an Abstract class called "**account**" with the following functionalities:

+default constructor

+parameterized constructor

+[p]name(can not be changer after creation)

+[p]ID(auto generated) (can not be changer after creation)

+[p]DOB (can not be changer after creation)

+[p]nominee(can be changed after creation)

+[p]double balance

+abstract bool deposit(double amount)//if the amount can be successfully deposited to the balance return true, else false

+ abstract bool withdraw(double amount)//if the amount can be successfully withdraw from the balance return true, else false

+ abstract void printAccount()//prints all the account info

Design a class called "**Debit**" from **“Account”**with the following functionalities:

+default constructor

+parameterized constructor

+bool deposit(double amount)//if the amount can be successfully deposited to the balance return true, else false

+bool withdraw(double amount)//if the amount can be successfully withdraw from the balance return true, else false

+void printAccount()//prints all the account info

\*\*\*debit account’s max balance is 100000 and daily transaction limit is 20000

Design a class called "**Credit**" from **“Account”** with the following functionalities:

+default constructor

+parameterized constructor

+bool deposit(double amount)//if the amount can be successfully deposited to the balance return true, else false

+bool withdraw(double amount)//if the amount can be successfully withdraw from the balance return true, else false

+void printAccount()//prints all the account info

\*\*\*Credit account’s min balance is -100000 and no upper limit,Cash withdraw limit is 50%, daily Cash withdraw limit is 20000

Design a class called "**Savings**" from **“Account”** with the following functionalities:

+default constructor

+parameterized constructor

+bool deposit(double amount)//if the amount can be successfully deposited to the balance return true, else false

+bool withdraw(double amount)//if the amount can be successfully withdraw from the balance return true, else false

+void printAccount()//prints all the account info

\*\*\*no limit

Design a class called "**bank**" with the following functionalities:

-array of ***account*** of size 100 [Map<String, base> m= new HashMap<String, base>(5);]

+void create\_account()//this method will take care of the user input for account creation

+void deposit(String ID, amount)//this method will take care of the user input for deposit an amount to a specific account

\*\*\*please generate error msg is account cannot be found or other

-void withdraw(String ID,ammount)//this method will take care of the user input for withdraw an amount from a specific account

\*\*\*please generate error msg is account cannot be found or other

-void print(String ID)//prints an specific account details

Design a class called “**BankIO”** which takes cares of the entire Console based input(Use Scanner class) and output