COMP41100

EXPLORING PROGRAMMING IN RUBY

PRACTICAL THREE

05th October 2013

Andrew Doyle
Student Number: 12252388

University College Dublin

INTRODUCTION

This section outlines the format of this submission; giving a short introduction to the program. It also includes assumptions made. The program will be explained in more detail in the Program Description section, where the author will guide the reader through the file **Nama.rb**, explaining each step (which will involve explaining the methods called from the files **Bank.rb** and **Developers.rb**).

An effort was made to provide somewhat accurate data for banks; to make the programme more meaningful. Data sources are included in the Bibliography. Data regarding developers is more difficult to find; therefore this data is fabricated.

How the programme works

The user is given various options, whereby they can:

1. Display information regarding banks
2. Display developer information
3. Check if banks are solvent (can check all banks at once or each bank separately)
    a. This step may involve loans being moved to nama
4. Check if developers are solvent (user can check all developers at once or each developer separately)
5. Check how much a developer owes altogether
6. Check how much a developer owes a bank
7. Implement changes dependant on bankruptcy status
    a. This step involves loan data and nama/solvency status being altered
8. Exit the programme

Steps 1 – 6 as outlined above involve the data being written to a file in the programme directory.

Assumptions:

Methods

owes_bank

1. 70% of Johnny Ronan`s loans are from a bank; namely Anglo.
2. 80% of Richard Barrett`s loans are from a bank; namely Allied Irish Banks.
3. 65% of Sean Mulryan`s loans are from a bank; namely Royal Bank of Scotland.
4. 90% of Bernard McNamara`s loans are from a bank; namely Bank of Ireland.
5. 95% of Liam Carroll`s loans are from a bank; namely Ulster Bank.

owes_amount

Returns the amount of money that the developer owes i.e. includes banks and other debts.

moves_loans_to_nama

If the amount of loans the bank has lent is greater than 1.2 times the amount of deposits then the **nama-status** is set to **true** and the **amount_of_loans** is decreased by 20%.

bankrupt_test

This method sets the **bankrupt** status to true if their total loans is greater than 25% higher than their net worth. It also subtracts the bank loans from the developer`s loans figures (which affects the developer`s solvency.

PROGRAM DESCRIPTION

The **Nama.rb** contains the class **Nama** which instantiates the **Bank** and **Developer** classes declared in the other files (which are referenced in **Nama.rb** using **require_relative**). The **Nama** class also makes use of the methods declared in the other classes.

The **Bank** and **Developer** classes are instantiated in the same manner as previous submissions using the **attr_accessor** and **initialize** methods. **Attr_accessor** is used to create a reader and writer for a given attribute and **initialize** allows initialization during instantiation of the class.

In the **Nama** class, instances of the **Bank** and **Developer** classes are created using the **new** keyword which are assigned to variables. This variables are then pushed into arrays using the **<<** operator:

```
banks = []

banks << aib << anglo << boi << rbs << ulster
```

A **while loop** is then set up to ask the user to carry out options until they choose to exit (which is referenced by **repeat**):

```
repeat = ''

while repeat != 0

    puts "\nWhat do you want to do?"
    puts "\n"
    puts "Check Bank Details                              | 0"
    puts "Check Developer Details                         | 1"
    puts "Check if a Bank is Solvent                      | 2"
    puts "  **Please Note - If the value of a banks loans is greater than 20% higher
than it`s deposits"
    puts "    20% of it`s loans will be moved to NAMA and NAMA status will be set to
true"
    puts "Check if a Developer is Solvent                 | 3"
    puts "Check how much a Developer owes (altogether)    | 4"
    puts "Check how much a Developer owes bank            | 5"
    puts "Perform bankruptcy tests and consequences       | 6"
    puts "Exit                                            | 7"

    userchoice = gets.chomp
    choice = userchoice.to_i

    case
```

The **case statement** carries out different task based on the user`s choice:

Case 0: Check Bank Details

Each instance of the **banks** array is output to the screen in a tabular format (made possible using **format %** - as completed for pratical 3, question 4). Furthermore, a text file is created using **File.open**. **Open** is a method from the **file** class. **Each** is used to iterate through the **banks** array.

Case 1: Check Developer Details

Developer details are output and written to a file in the same manner as **Case 0**. **Do** is used to iterate through the **developers** array when outputting the information to the user.

Case 2: Check Bank Solvency

The user is given the choice to check all banks, or to check each bank individually. A nested **case statement** carries out appropriate actions based on the user`s choice. Before the case statement, the **bank_is_solvent** and **moves_loans_to_nama** methods are called on each bank. The methods were declared in the **Bank** class from the **Bank.rb** file:

```
def bank_is_solvent
  @amount_of_deposits = amount_of_deposits.to_i
  @amount_of_loans = amount_of_loans.to_i
  if amount_of_deposits > amount_of_loans then
    return true
  else
    return false
  end
end
```

The above method returns **true** for each bank if the amount of deposits are greater than the amount of loans. The attributes are referenced using **@**. **To_i** is used to convert the strings to integers.

```
def moves_loans_to_nama
    @nama_status = nama_status
    @amount_of_loans = amount_of_loans.to_i
    @amount_of_deposits = amount_of_deposits.to_i
  if amount_of_loans > (1.2 * amount_of_deposits) then
    @nama_status = true
    float_amount_of_loans = amount_of_loans * 0.8
    @amount_of_loans = float_amount_of_loans.to_i
  end

end
```

The above method sets the **nama-status** as true if the amount of loans on the bank`s books are greater than 20% higher than their deposits. In the aforementioned scenario the amount of loans is also reduced by 20%.

In the nested **case-statement** a message is displayed stating whether or not the chosen bank is solvent or solvent; this is determined by whether or not **bank_is_solvent** returns **true** (all bank information is displayed in **Case 0**). Information regarding bank solvency is also written to a file in **Case 0**.

Case 3: Check Developer Solvency

Developer solvency information is displayed and written to a file in the same manner as Case 2 (determined by **is_developer_solvent?** which is declared in the **Developers** class).

```
def is_developer_solvent?
   net_worth = @net_worth
   loans = @loans
   net = net_worth.to_i
   loans = loans.to_i
   if net > loans then
     return true
   else
     return false
   end
end
```

The above method returns **true** if the developer`s net worth is greater than the value of his loans.

Case 4 & 5: Checking how much the developer owes (In total) and how much they owe the bank

Case 4 and 5 display information to the screen and write information to a file in the same manner as previous steps.

Case 6: Performing bankruptcy checks and implementing consequences

Case 6 implements the **bankrupt_test** method which was declared in the **Developers** class:

```
  def bankrupt_test
    @net_worth = net_worth
    @loans = loans
    @bankrupt = bankrupt
    @name = name

    net_worth_int = net_worth.to_i
    loans_total_int = loans.to_i
    dev_name = name.chomp
    if loans_total_int > (net_worth_int * 1.25)then
      @bankrupt = true

    end

  end
```

As shown above, a developer is declared bankrupt if the value of their loans exceeds their net worth by more than 25%. In terms of **Case 6**, if the user choose to proceed, the **bankrupt_test** method is called on each developer. If the developer is bankrupt their **loans** figure is reduced (by the amount they owe banks – other debts remain). The relevant bank`s loans are also reduced whilst the bank`s nama-status is also set to **true**.

For example, the Developer Bernard McNamara is affected by the above. The authoer recommends that for a more optimum program the following code is implemented in a more dynamic/efficient/less code heavy manner:

```
bernard_mcnamara.bankrupt_test

if bernard_mcnamara.bankrupt == true then
  banks_loans = rbs.amount_of_loans.to_i
  bernards_loans = bernard_mcnamara.loans.to_i
  new_banks_loans = ( banks_loans -= (bernards_loans * 0.90))
  rbs.amount_of_loans = new_banks_loans.to_i
  new_loan_figure = bernards_loans * 0.10
  bernard_mcnamara.loans = new_loan_figure.to_i
  rbs.nama_status = true
end
```

If the user does not want to implement the bankrupt consequences, they may exit **Case 6** using the **SystemExit** command:

```
#abandon bankruptcy chances and have the option to continue or exit
when choice == 1 then
   SystemExit
```

Case 7: Exit

```
when choice == 7 then
  puts "\nThanks..Have a great day!"
  exit + 2
```

The user has the option to exit the program altogether, made possible by **exit + 2**.

The user then has the opportunity to carry out **Case 0 – 6** once again (this has the benefit of being able to easily view the changes made):

```
puts "\nDo you want to check information again?"
puts "Exit          | 0"
puts "Check Again   | 1"
repeat = gets.chomp
repeat = repeat.to_i #users choice on whether to repeat program - this is the condition for the loop
```

APPENDIX A – SAMPLE OUTPUT FROM CONSOLE VIEW AND FILES

Figure 1 – Checking Bank Details

```
What do you want to do?

Check Bank Details                          | 0
Check Developer Details                     | 1
Check if a Bank is Solvent                  | 2
  **Please Note - If the value of a banks loans is greater than 20% higher than it`s deposits
    20% of it`s loans will be moved to NAMA and NAMA status will be set to true
Check if a Developer is Solvent             | 3
Check how much a Developer owes (altogether) | 4
Check how much a Developer owes bank        | 5
Perform bankruptcy tests and consequences   | 6
Exit                                        | 7
0

-------------------- BANK DETAILS ------------------------

Bank Name              Number of Employees Amount of Deposits Amount of Loans Nama Status
Allied Irish Banks     13000               102718000000       75886000000     true
Anglo                  850                 59000000000        58000000000     true
Bank of Ireland        12000               104000000000       103000000000    true
Royal Bank of Scotland 141000              580000000000       610000000000    false
Ulster Bank            3250                27000000000        39000000000     false

The above information has been written to bank_details.txt in the directory

Do you want to check information again?
Exit          | 0
Check Again   | 1
1
```

Figure 2 – Checking Developer Details

```
What do you want to do?

Check Bank Details                          | 0
Check Developer Details                     | 1
Check if a Bank is Solvent                  | 2
  **Please Note - If the value of a banks loans is greater than 20% higher than it`s deposits
    20% of it`s loans will be moved to NAMA and NAMA status will be set to true
Check if a Developer is Solvent             | 3
Check how much a Developer owes (altogether) | 4
Check how much a Developer owes bank        | 5
Perform bankruptcy tests and consequences   | 6
Exit                                        | 7
1

-------------------- DEVELOPER DETAILS ------------------------

Developers Name  Net Worth Loans    Bank   Bankrupt
Johnny Ronan     25000000  50000000 Anglo
Richard Barrett  15000000  10000000 AIB
Sean Mulryan     12000000  14000000 BOI
Bernard McNamara 30000000  40000000 RBS
Liam Carroll     22000000  35000000 ULSTER

Do you want to check information again?
Exit          | 0
Check Again   | 1
1
```

Figure 3 – Checking Bank Solvency

```
What do you want to do?

Check Bank Details                          | 0
Check Developer Details                     | 1
Check if a Bank is Solvent                  | 2
  **Please Note - If the value of a banks loans is greater than 20% higher than it`s deposits
    20% of it`s loans will be moved to NAMA and NAMA status will be set to true
Check if a Developer is Solvent             | 3
Check how much a Developer owes (altogether) | 4
Check how much a Developer owes bank        | 5
Perform bankruptcy tests and consequences   | 6
Exit                                        | 7
2

CHECKING BANK SOLVENCY

Which Bank do you want to Check?

Check All Banks                     | 0
AIB                                 | 1
ANGLO                               | 2
BOI                                 | 3
RBS                                 | 4
ULSTER                              | 5
0
Allied Irish Banks is solvent
Anglo is solvent
Bank of Ireland is solvent
Royal Bank of Scotland is not solvent
Ulster Bank is not solvent

Bank solvency details have been written to bank_solvency_status.txt
```

Figure 4 – Checking Developer Solvency

```
What do you want to do?

Check Bank Details                          | 0
Check Developer Details                     | 1
Check if a Bank is Solvent                  | 2
  **Please Note - If the value of a banks loans is greater than 20% higher than it`s deposits
    20% of it`s loans will be moved to NAMA and NAMA status will be set to true
Check if a Developer is Solvent             | 3
Check how much a Developer owes (altogether) | 4
Check how much a Developer owes bank        | 5
Perform bankruptcy tests and consequences   | 6
Exit                                        | 7
3

CHECKING DEVELOPER SOLVENCY

Which Developer do you want to Check?

Check All Developers            | 0
Johnny Ronan                    | 1
Richard Barrett                 | 2
Sean Mulryan                    | 3
Bernard McNamara                | 4
Liam Carroll                    | 5
0
Johnny Ronan is not solvent
Richard Barrett is solvent
Sean Mulryan is not solvent
Bernard McNamara is not solvent
Liam Carroll is not solvent

Developer solvency details have been written to developer_solvency_status.txt
```

Figure 5 – Checking what Developer Owes Altogether

```
What do you want to do?

Check Bank Details                          | 0
Check Developer Details                     | 1
Check if a Bank is Solvent                  | 2
  **Please Note - If the value of a banks loans is greater than 20% higher than it`s deposits
    20% of it`s loans will be moved to NAMA and NAMA status will be set to true
Check if a Developer is Solvent             | 3
Check how much a Developer owes (altogether) | 4
Check how much a Developer owes bank        | 5
Perform bankruptcy tests and consequences   | 6
Exit                                        | 7
4

CHECKING HOW MUCH DEVELOPER OWES (ALTOGETHER)

Which Developer do you want to Check?

Check All Developers                | 0
Johnny Ronan                        | 1
Richard Barrett                     | 2
Sean Mulryan                        | 3
Bernard McNamara                    | 4
Liam Carroll                        | 5
0
Johnny Ronan owes €50000000 altogether
Richard Barrett owes €10000000 altogether
Sean Mulryan owes €14000000 altogether
Bernard McNamara owes €40000000 altogether
Liam Carroll owes €35000000 altogether

The above details have been written to developer_owes_money.txt
```

Figure 6 – Checking what Developer Owes Banks

```
What do you want to do?

Check Bank Details                          | 0
Check Developer Details                     | 1
Check if a Bank is Solvent                  | 2
  **Please Note - If the value of a banks loans is greater than 20% higher than it`s deposits
    20% of it`s loans will be moved to NAMA and NAMA status will be set to true
Check if a Developer is Solvent             | 3
Check how much a Developer owes (altogether) | 4
Check how much a Developer owes bank        | 5
Perform bankruptcy tests and consequences   | 6
Exit                                        | 7
5

CHECKING HOW MUCH DEVELOPER OWES BANKS

Which Developer do you want to Check?

Check All Developers                | 0
Johnny Ronan                        | 1
Richard Barrett                     | 2
Sean Mulryan                        | 3
Bernard McNamara                    | 4
Liam Carroll                        | 5
0
Johnny Ronan owes the banks €35000000 altogether
Richard Barrett owes the banks €8000000 altogether
Sean Mulryan owes the banks €9100000 altogether
Bernard McNamara owes the banks €36000000 altogether
Liam Carroll owes the banks €33250000 altogether

The above details have been written to developer_owes_banks.txt
```

Figure 7 – Implementing Bankruptcy Changes

```
What do you want to do?

Check Bank Details                          | 0
Check Developer Details                     | 1
Check if a Bank is Solvent                  | 2
  **Please Note - If the value of a banks loans is greater than 20% higher than it`s deposits
    20% of it`s loans will be moved to NAMA and NAMA status will be set to true
Check if a Developer is Solvent             | 3
Check how much a Developer owes (altogether) | 4
Check how much a Developer owes bank        | 5
Perform bankruptcy tests and consequences   | 6
Exit                                        | 7
6

PERFORMING BANKRUPTCY TEST (HAS MAJOR CONSEQUENCES

Please note that this action has major consequences:
 # The developer will be declared bankrupt if his loans are more than 25% higher than his net worth
 # If the developer is declared bankrupt:
   - This will reduce the amount of money they owe - which may make them solvent
   - The relevant bank`s loan book will be reduced by the relevant amount, and NAMA status will be set to true

Are you sure that you wish to perform this task?

Yes - perform bankruptcy test and implement consequences | 0
No                                                       | 1
0

Do you want to check information again?
Exit          | 0
Check Again   | 1
```

Figure 8 - Exiting

```
What do you want to do?

Check Bank Details                          | 0
Check Developer Details                     | 1
Check if a Bank is Solvent                  | 2
  **Please Note - If the value of a banks loans is greater than 20% higher than it`s deposits
    20% of it`s loans will be moved to NAMA and NAMA status will be set to true
Check if a Developer is Solvent             | 3
Check how much a Developer owes (altogether) | 4
Check how much a Developer owes bank        | 5
Perform bankruptcy tests and consequences   | 6
Exit                                        | 7
7

Thanks..Have a great day!
```

Figure 9 – Files after first iteration through program

bank_details.txt - Notepad

File Edit Format View Help

Bank Name	Number of Employees	Amount of Deposits	Amount of Loans	Nama Status
Allied Irish Banks	13000	102718000000	75886000000	true
Anglo	850	59000000000	58000000000	true
Bank of Ireland	12000	104000000000	103000000000	true
Royal Bank of Scotland	141000	580000000000	610000000000	false
Ulster Bank	3250	27000000000	39000000000	false

Ln 1, Col 1

developer_details.txt - Notepad

File Edit Format View Help

Developers Name	Net Worth	Loans	Bank	Bankrupt
Johnny Ronan	25000000	50000000	Anglo	
Richard Barrett	15000000	10000000	AIB	
Sean Mulryan	12000000	14000000	BOI	
Bernard McNamara	30000000	40000000	RBS	
Liam Carroll	22000000	35000000	ULSTER	

Ln 1, Col 1

bank_solvency_status.txt - Notepad

File Edit Format View Help

```
Allied Irish Banks is solvent
Anglo is solvent
Bank of Ireland is solvent
Royal Bank of Scotland is not solvent
Ulster Bank is not solvent
```

Ln 1, Col 1

developer_solvency_status.txt - Notepad

File Edit Format View Help

```
Johnny Ronan is not solvent
Richard Barrett is solvent
Sean Mulryan is not solvent
Bernard McNamara is not solvent
Liam Carroll is not solvent
```

Ln 1, Col 1

developer_owes_money.txt - Notepad

File Edit Format View Help

```
Johnny Ronan owes €50000000 altogether
Richard Barrett owes €10000000 altogether
Sean Mulryan owes €14000000 altogether
Bernard McNamara owes €40000000 altogether
Liam Carroll owes €35000000 altogether
```

Ln 1, Col 1

developer_owes_banks.txt - Notepad

File Edit Format View Help

```
Johnny Ronan owes the banks €35000000 altogether
Richard Barrett owes the banks €8000000 altogether
Sean Mulryan owes the banks €9100000 altogether
Bernard McNamara owes the banks €36000000 altogether
Liam Carroll owes the banks €33250000 altogether
```

Ln 1, Col 1

assed during instantiation as

the class are then instantiated

Figure 10 – Files after second iteration through program

bank_details.txt - Notepad

File Edit Format View Help

Bank Name	Number of Employees	Amount of Deposits	Amount of Loans	Nama Status
Allied Irish Banks	13000	102718000000	75886000000	true
Anglo	850	59000000000	57965000000	true
Bank of Ireland	12000	104000000000	103000000000	true
Royal Bank of Scotland	141000	580000000000	609964000000	true
Ulster Bank	3250	27000000000	31166750000	true

Ln 1, Col 1

developer_details.txt - Notepad

File Edit Format View Help

Developers Name	Net Worth	Loans	Bank	Bankrupt
Johnny Ronan	25000000	15000000	Anglo	true
Richard Barrett	15000000	10000000	AIB	
Sean Mulryan	12000000	14000000	BOI	
Bernard McNamara	30000000	4000000	RBS	true
Liam Carroll	22000000	1750000	ULSTER	true

Ln 1, Col 1

bank_solvency_status.txt - Not...

File Edit Format View Help

```
Allied Irish Banks is solvent
Anglo is solvent
Bank of Ireland is solvent
Royal Bank of Scotland is not solvent
Ulster Bank is not solvent
```

Ln 1, Col 1

developer_solvency_status.txt - Notepad

File Edit Format View Help

```
Johnny Ronan is solvent
Richard Barrett is solvent
Sean Mulryan is not solvent
Bernard McNamara is solvent
Liam Carroll is solvent
```

Ln 1, Col 1

developer_owes_money.txt - Notepad

File Edit Format View Help

```
Johnny Ronan owes €15000000 altogether
Richard Barrett owes €10000000 altogether
Sean Mulryan owes €14000000 altogether
Bernard McNamara owes €4000000 altogether
Liam Carroll owes €1750000 altogether
```

Ln 1, Col 1

developer_owes_banks.txt - Notepad

File Edit Format View Help

```
Johnny Ronan owes the banks €10500000 altogether
Richard Barrett owes the banks €8000000 altogether
Sean Mulryan owes the banks €9100000 altogether
Bernard McNamara owes the banks €3600000 altogether
Liam Carroll owes the banks €1662500 altogether
```

_solvency_status.txt

ogether

BIBLIOGRAPHY

1. AIB (2013). 2012 Annual Financial Report. *Aib.ie*. Retrieved 04/10/2013, from http://www.aib.ie/servlet/ContentServer?pagename=AIB_Investor_Relations/AIB_Download/aib_d_download&c=AIB_Download&cid=1367264523254&channel=IRFP

2. AIB (n.d). Developments in Recent Years. *Aib.ie*. Retrieved 04/10/2013, from http://www.aib.ie/servlet/ContentServer?pagename=AIB_Investor_Relations/Miscellaneous/ir_article_printer&c=AIB_Article&cid=1308216854130&channel=IRHP

3. Irish Bank Resolution Corporation (2012). Interim Report, Six Months ended 30 June 2012. Retrieved 04/10/2013, from http://www.ibrc.ie/About_us/Financial_information/Latest_interim_report/Interim_Report_2012.pdf

4. Irish Times (2013). Irish Government plays down UK proposal to swap Ulster Bank for Nama`s British loans. *Irishtimes.*com. Retrieved 04/10/2013 from http://www.irishtimes.com/business/sectors/financial-services/irish-government-plays-down-uk-proposal-to-swap-ulster-bank-for-nama-s-british-loans-1.1417216

5. Irish Times (2013). Is there a bit more bang left in the bearded buccaneer? Retrieved 04/10/2013 from http://www.independent.ie/business/commercial-property/is-there-a-bit-more-bang-left-in-the-bearded-buccaneer-29618199.html.

6. New Ireland Assurance (2013). Bank of Ireland Annual Report, For the year ended 31 December 2012. *Newireland.ie*. Retrieved 04/10/2013 from http://www.newireland.ie/docs/general-documents/boi-annual-report-2012.pdf

7. RBS Group (2013). RBS Group annual report and accounts year ending 31 December 2012. *Investors.rbs.com*. Retrieved 04/10/2013 from http://www.investors.rbs.com/download/report/Annual_Report_2012.pdf

8. Ulster Bank (2013). Report of Directors and Financial Statements. Retrieved 04/10/2013 from http://group.ulsterbank.com/content/downloads/UBIL_2012_accounts.pdf

LIST OF FIGURES