Web Application and Development: Group Assignment Manual

CS1524 / CS1525

March 8, 2020

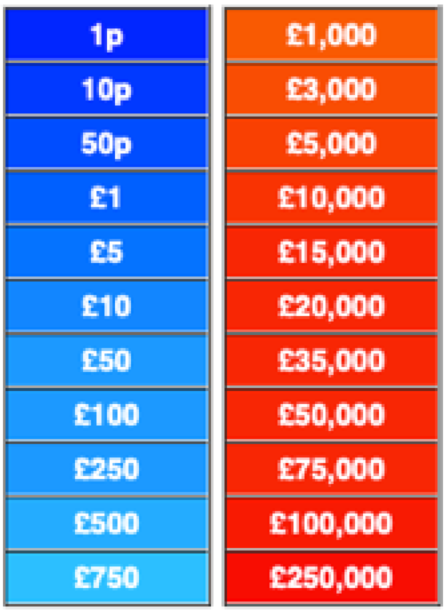
Andy Szalma, Antonia Duncan, Cammy Begg, Laura McKenna, Matey Krastev

Contents

1 Introduction

1.1 Preface: Understanding the Game

The purpose of this manual is to give a detailed insight into the creation process of our interpretation of the game, Deal-or-no-Deal. This was a UK game show that was adapted from a show in the Netherlands. The game consists of a user choosing a box numbered from 1-22. These boxes contain different amounts of cash, as shown below.



1p	£1,000
10p	£3,000
50p	£5,000
£1	£10,000
£5	£15,000
£10	£20,000
£50	£35,000
£100	£50,000
£250	£75,000
£500	£100,000
£750	£250,000

The user's aim is to win the maximum amount of cash as possible by potentially gambling for offers provided by the designated "banker".

There are different rounds in which the user choses a box value 1-22 in their chosen order. The value within the box is revealed and is scored off on the list of values the user could win. The banker makes offers between rounds depending on the values of the boxes already opened.

If the boxes opened were in the blue range of values above, the banker would most likely offer a higher amount, however, if the boxes opened were in the red range, the banker would most likely offer a lower amount.

For greater insight of the game, refer to the following link:
https://en.wikipedia.org/wiki/Deal_or_No_Deal_(British_game_show)

The game show stopped in 2016, but this guide and code aims to capture the essence of the game.

The first section provided is an addition to the manual. It provides insight towards the purpose and instructions on how to use the manual. It also briefly discusses the team contribution through the process of writing the code and manual.

1.2 About the Manual

The given requirements from the Assessment Template (Aberdeen_Assessment_template2019-20_CA2.docx), specifies to complete the tasks outlined in the spec file within the zip folder provided.

Tests are an important part of the process of developing programs such as games. They help ensure the user is satisfied with the result as the completion of tasks individually helps eliminate errors. We followed the TDD process as specified in the lecture 12 slides (Testing With Ruby) as well as the provided Template for TDD in the wad_dond_gen_01.rb, both by Nigel Beacham.

1.3 How to Use the Test Manual

The layout of the test manual is described in the table of contents. The first section is as described in the preface.

The second section goes through the prerequisites such as the test assumptions, system requirements, installation process and run instructions for the tasks. The test assumptions will describe any assumptions made while writing the code. These specifications are important to read/consider in order to further the understanding of interpreting and reading the code. Following the assumptions, there are system requirements. These are requirements that you will need in order to view and run the tasks in the terminal using ruby. It will specify the version of ruby required and any gems that have been used. Furthermore, this section contains the instructions in order to download the files and extract them. The final subsection will consist of the instructions on how to run the code to see whether the tasks are completed.

The next section describes the functionality of the game. This entails some of the key parts of understanding the game. This will show the detailed design using pseudocode, go through a step-by-step account of the different tasks and how they have passed.

1.4 Team Collaboration

We divided the tasks, provided in the assessment template, in a way such that everyone was doing an equal amount in an area they felt comfortable. We all decided, for greater understanding, to each individually complete the 38 tasks required. Once that was completed we compared our code and moved onto the creation of the manual. The manual was written as a collective and was split equally as noted in the attached team contribution form. We arranged to work collectively in our practicals and meet when necessary. We would still communicate through messenger when meeting up was not possible.

Furthermore, similar to the first assessment we used github as an efficient tool in order to share our code in an effective and reliable manner. While merging the code, we used pull requests as it was a simple and effective way to combine our ideas. These tools enabled us to communicate efficiently and work effectively as a group in order to write and share readable and reusable code.

2 Prerequisites

2.1 Test Assumptions

- Your machine satisfies the system requirements presented below, and you have followed correctly the run instructions.
- The tasks requested were interpreted and completed as intended.

With regard to the game

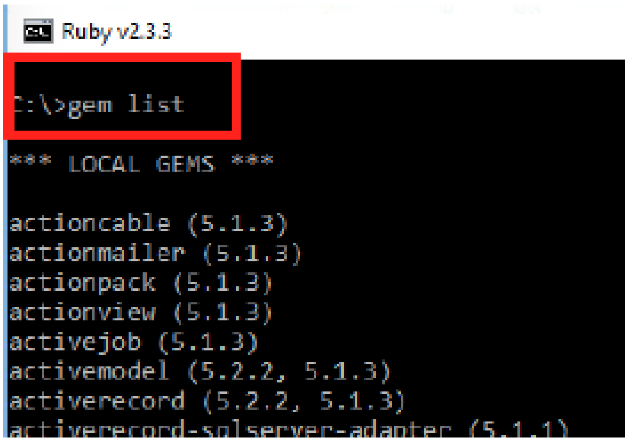
- The values in the boxes are determined based on the British version of the show, other amounts for the values in the boxes can also be assumed.
- The user chooses one box and then the banker makes an offer (not multiple turns), when in the real game show it is not set and the banker can be more flexible.
- The value that the banker offers is written as the following: value = value/2.
  - Calculations in reality are much more complex, however, for simplicity we chose the above idea.

2.2 System Requirements

System Requirements for running the game:

- Ruby 2.3.3 or later installed on the machine. There is no other alternative to using ruby for the files, therefore, this is a necessity. If any additional help is required for the installation of ruby, visit the following webpage: https://www.ruby-lang.org/en/
- An additional gem will be required, called “rspec”. In order to download this visit: https://rubygems.org/pages/download
- Operating system: Windows 10. This is the operating system used and therefore in the manual it was only discussed with windows 10.

You can check the gems you already have on your ruby application by typing “gem list” into the command window, as shown below.

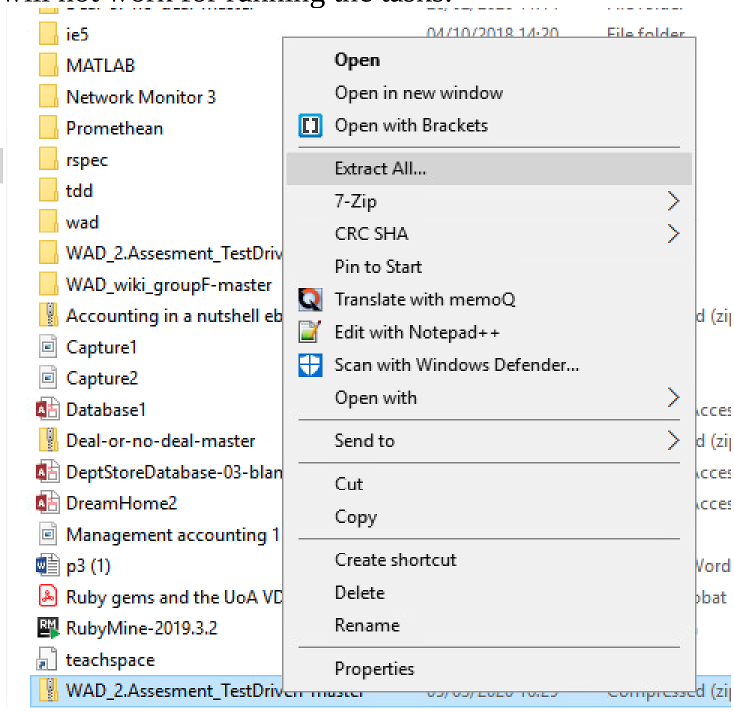


```
:\>gem list

*** LOCAL GEMS ***

actioncable (5.1.3)
actionmailer (5.1.3)
actionpack (5.1.3)
actionview (5.1.3)
activejob (5.1.3)
activemodel (5.2.2, 5.1.3)
activerecord (5.2.2, 5.1.3)
activerecord-sqlserver-adapter (5.1.1)
```

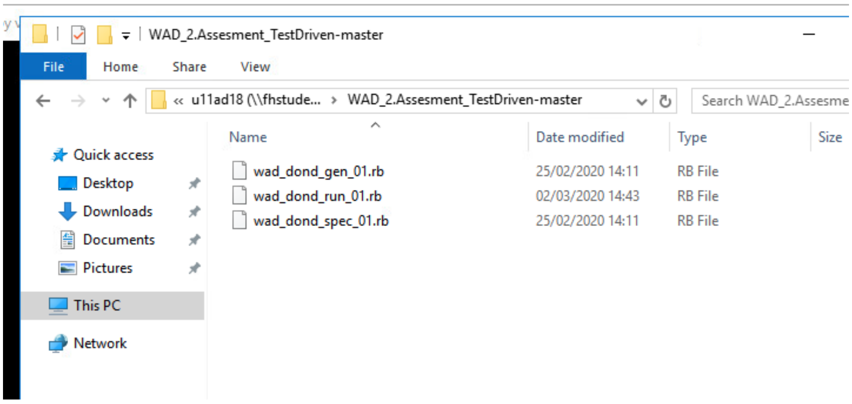
2.3 Installation Details

In order to run the tasks, all of the system requirements are a necessity. Then continue to follow the instructions on how to install the folder.

1. Download the zip file and extract all from the folder. This will save the folder as a standard folder as opposed to a compressed, zip folder. Keeping it as a zipped file will not work for running the tasks.



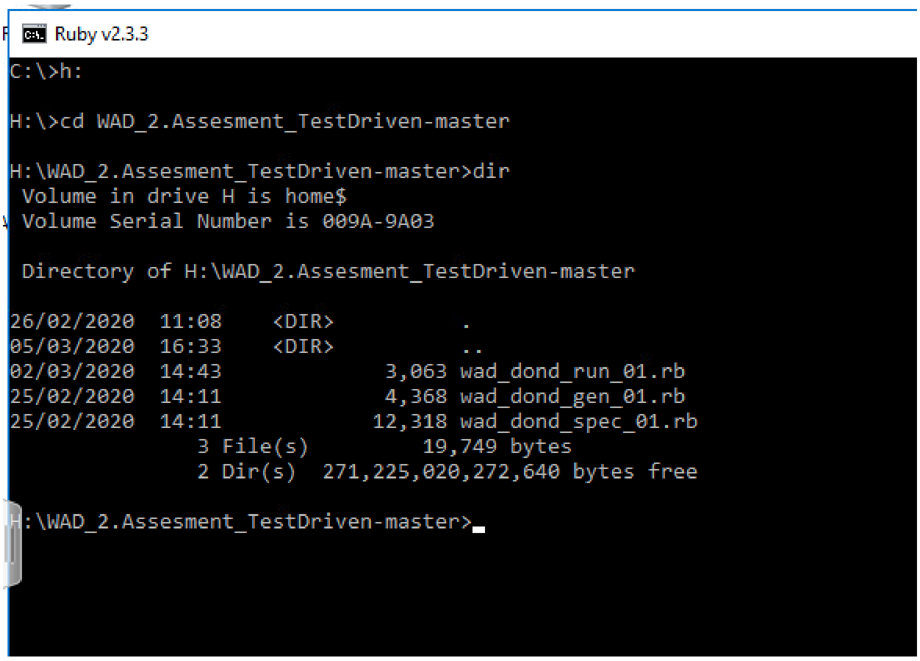
2. Once all of the files have been extracted and the folder is saved, recall the location where it is saved. For example, the above folder has been extracted to the h: drive of this pc. The below is an example of the pathway.



Name	Date modified	Type	Size
wad_dond_gen_01.rb	25/02/2020 14:11	RB File	
wad_dond_run_01.rb	02/03/2020 14:43	RB File	
wad_dond_spec_01.rb	25/02/2020 14:11	RB File	

3. The location of the folder and the files within it can be checked using the following ruby commands in the ruby terminal:

   1. h:
   2. cd WAD_2.Assessment_TestDriven-master



```
C:\>h:

H:\>cd WAD_2.Assesment_TestDriven-master

H:\WAD_2.Assesment_TestDriven-master>dir
 Volume in drive H is home$
 Volume Serial Number is 009A-9A03

 Directory of H:\WAD_2.Assesment_TestDriven-master

26/02/2020  11:08    <DIR>          .
05/03/2020  16:33    <DIR>          ..
02/03/2020  14:43             3,063 wad_dond_run_01.rb
25/02/2020  14:11             4,368 wad_dond_gen_01.rb
25/02/2020  14:11            12,318 wad_dond_spec_01.rb
               3 File(s)         19,749 bytes
               2 Dir(s)  271,225,020,272,640 bytes free

H:\WAD_2.Assesment_TestDriven-master>
```

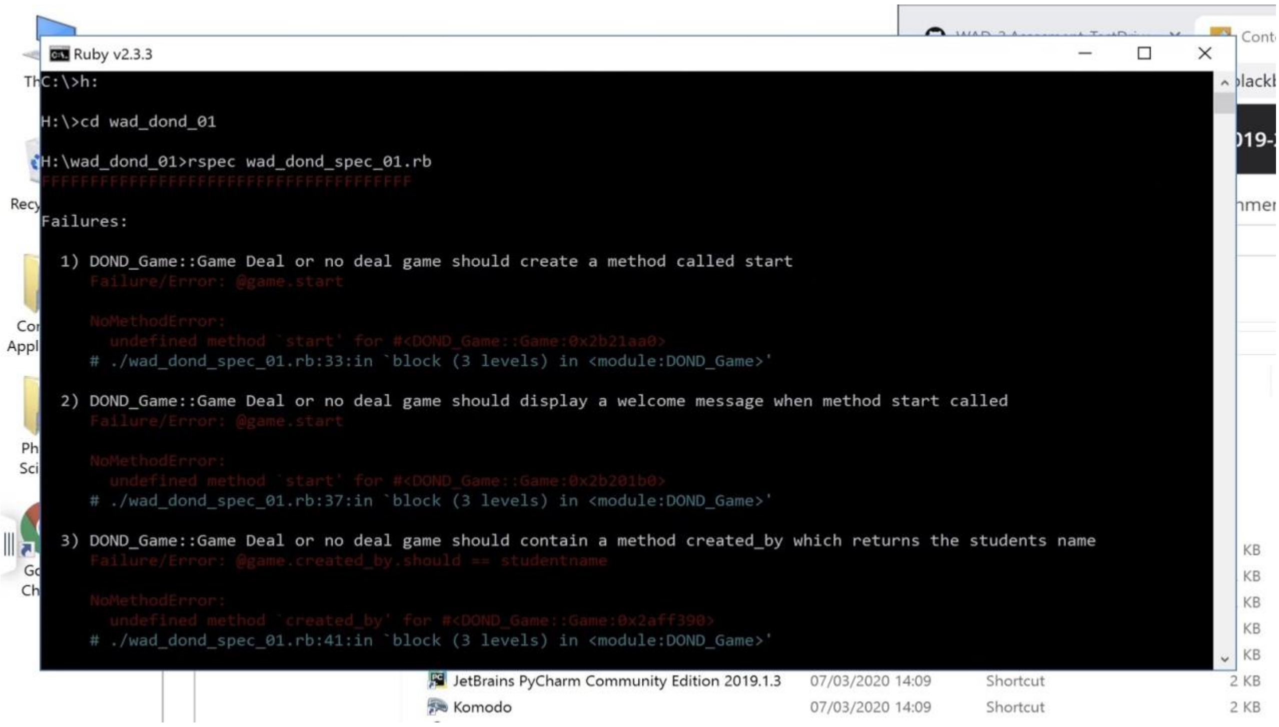
In the image above it also uses the command “dir” in order to assure that the same files from the zip folder are in the unzipped folder.

4. Once located in the corresponding folder (in the above example, the WAD_2.Assessment_TestDriven-master folder), the commands to check whether the tests have been implemented can be run.

2.4 Run Instructions

In order to test whether the tasks have been completed, it is specified in the template provided to run the following command: rspec wad_dond_gen.rb . Note the folder you are located in. Recall you have to re-enter the pathway if the ruby application is closed. In this example the folder is saved in the h: drive, and in the folder wad_dond_01 in the first case, and the commands are corresponding to those names. Please note that this is an example, and corresponding adjustments to the folder name and location may be different amongst different users.

The initial test was completed to assure that all 38 tasks failed, as shown below.

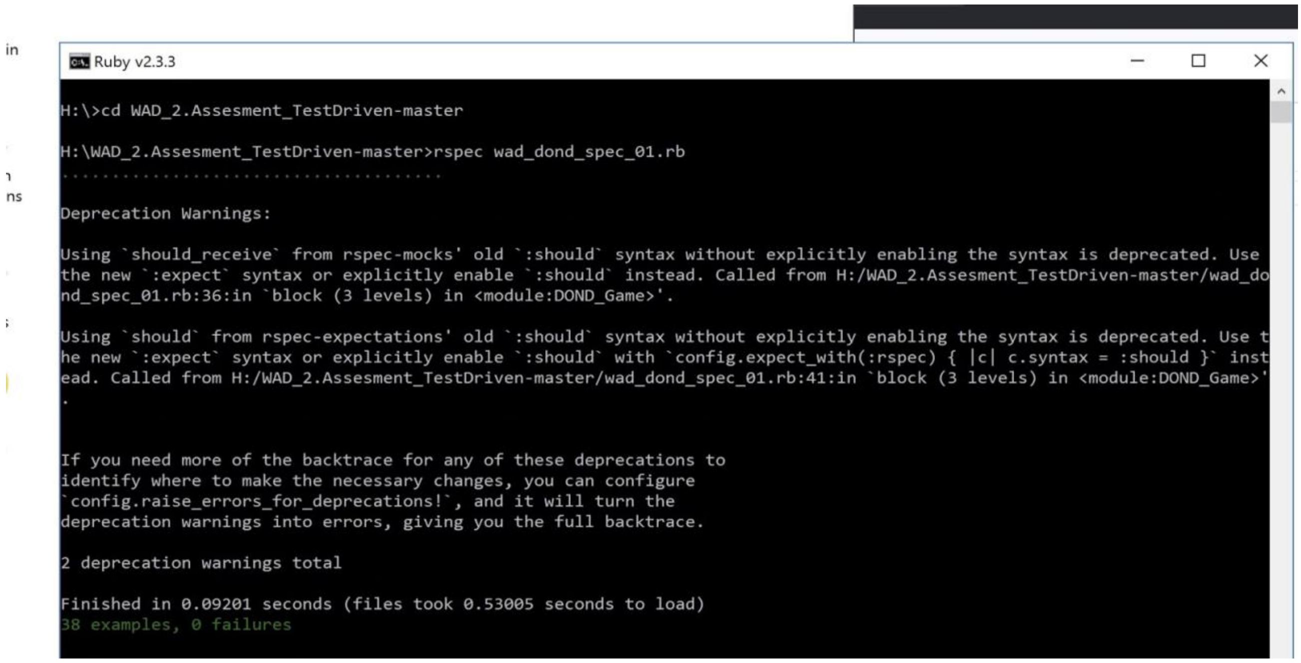


```
C:\>h:

H:\>cd wad_dond_01

H:\wad_dond_01>rspec wad_dond_spec_01.rb
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Failures:

  1) DOND_Game::Game Deal or no deal game should create a method called start
     Failure/Error: @game.start

     NoMethodError:
       undefined method `start' for #<DOND_Game::Game:0x2b21aa0>
     # ./wad_dond_spec_01.rb:33:in `block (3 levels) in <module:DOND_Game>'

  2) DOND_Game::Game Deal or no deal game should display a welcome message when method start called
     Failure/Error: @game.start

     NoMethodError:
       undefined method `start' for #<DOND_Game::Game:0x2b201b0>
     # ./wad_dond_spec_01.rb:37:in `block (3 levels) in <module:DOND_Game>'

  3) DOND_Game::Game Deal or no deal game should contain a method created_by which returns the students name
     Failure/Error: @game.created_by.should == studentname

     NoMethodError:
       undefined method `created_by' for #<DOND_Game::Game:0x2aff390>
     # ./wad_dond_spec_01.rb:41:in `block (3 levels) in <module:DOND_Game>'
```

JetBrains PyCharm Community Edition 2019.1.3 07/03/2020 14:09 Shortcut 2 KB
Komodo 07/03/2020 14:09 Shortcut 2 KB

The above screenshot shows the failures; there were 38 failures.

Once the tasks had been completed, re-running this then showed the following result:



```
H:\>cd WAD_2.Assesment_TestDriven-master

H:\WAD_2.Assesment_TestDriven-master>rspec wad_dond_spec_01.rb
......................................

Deprecation Warnings:

Using `should_receive` from rspec-mocks' old `:should` syntax without explicitly enabling the syntax is deprecated. Use
the new `:expect` syntax or explicitly enable `:should` instead. Called from H:/WAD_2.Assesment_TestDriven-master/wad_do
nd_spec_01.rb:36:in `block (3 levels) in <module:DOND_Game>'.

Using `should` from rspec-expectations' old `:should` syntax without explicitly enabling the syntax is deprecated. Use t
he new `:expect` syntax or explicitly enable `:should` with `config.expect_with(:rspec) { |c| c.syntax = :should }` inst
ead. Called from H:/WAD_2.Assesment_TestDriven-master/wad_dond_spec_01.rb:41:in `block (3 levels) in <module:DOND_Game>'
.


If you need more of the backtrace for any of these deprecations to
identify where to make the necessary changes, you can configure
`config.raise_errors_for_deprecations!`, and it will turn the
deprecation warnings into errors, giving you the full backtrace.

2 deprecation warnings total

Finished in 0.09201 seconds (files took 0.53005 seconds to load)
38 examples, 0 failures
```

The green text at the bottom displays the message “38 examples, 0 failures”. This shows that all of the tasks have been completed as specified in the spec file, and there are no failures.

3 Functionality

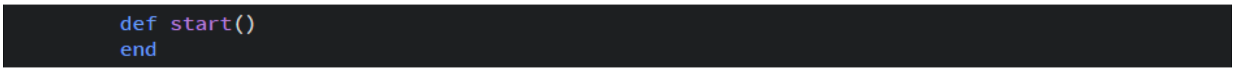
3.1 Detailed Design (pseudocode)

```
Start playing
    While not stop or player wins
        Output ‘Welcome to Deal or No Deal!’
        Output ‘Designed by’ + created_by
        Output ‘Student ID’ + student_id
        Output ‘Starting game…’
        Display menu
            Input user inputs the value from menu
                If input = 9
                    End game
                If input = 3
                    Output ‘Game analysis…’
                    Repeat
                        If opened box = 0
                            Status = closed
                        Else
                            Status = open
                    Output ‘[number of box]: [status (open/closed)]’
                    Until all box status’ have been revealed
                If input = 2 [resetgame called]
                    Output ‘New game…’
                    Set object variables to default
                    Assign values to boxes
                    Output ‘Enter the number of the box you wish to
                        keep’
                    Input user inputs a number of the box
                        If user answer<23 and user answer>0
                            game start
                        Else
                            Output ‘Error: box number must
                                    be 1 to 22’
                    Repeat
                        Output ‘Enter the number of the box you
                            wish to open’
                        Input user inputs a number
                            If input number has status closed
                                Display value
                            Then
                                Output ‘Banker offers you
                                    for your chosen box: [value]’
                                Update opened boxes
                                Display total closed
                                    boxes
                            Else
                                Display error
                    Until all boxes opened or offer accepted
                    Output ‘...game finished’
Finish
```

3.2 Account of Tests Implemented

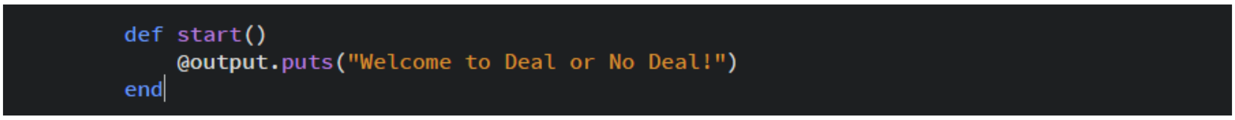
Note The tasks that were required to be completed are specified below (total of 38). The tasks were specified on the spec file (wad_dond_spec_01.rb). Below is a list of the tests that were required to be implemented.

1. Should create a method called start.



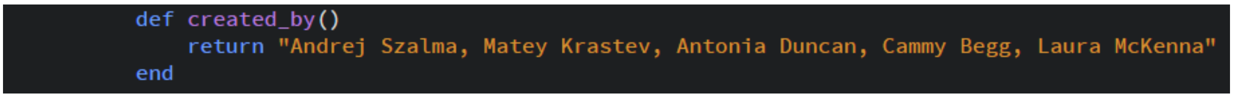
```
def start()
end
```

2. Should display a welcome message when method start called



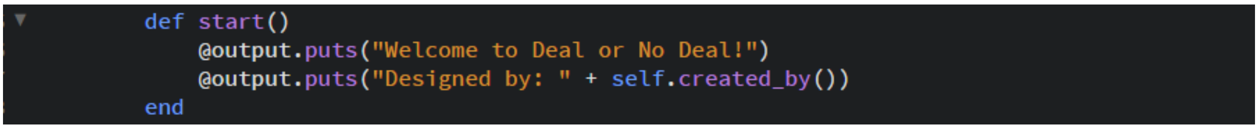
```
def start()
    @output.puts("Welcome to Deal or No Deal!")
end
```

3. Should contain a method created_by which returns the students name



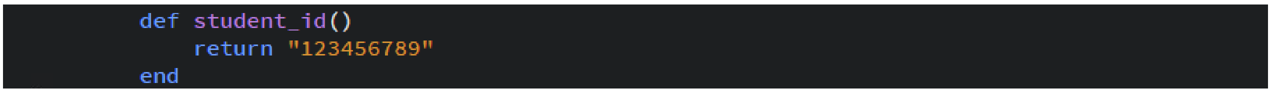
```
def created_by()
    return "Andrej Szalma, Matey Krastev, Antonia Duncan, Cammy Begg, Laura McKenna"
end
```

4. Should display a message showing who designed the game when the method start called



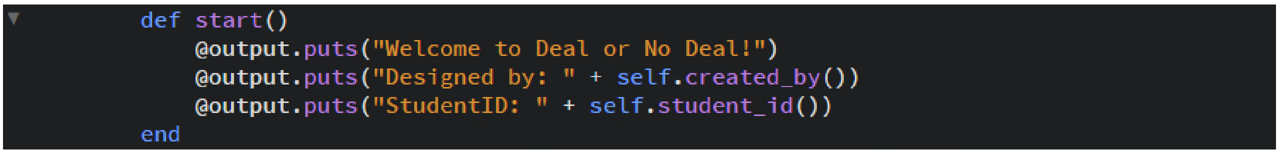
```
def start()
    @output.puts("Welcome to Deal or No Deal!")
    @output.puts("Designed by: " + self.created_by())
end
```

5. Should contain a method student_id which returns the students ID number



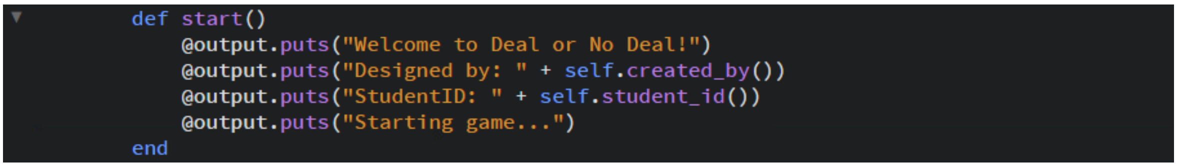
```
def student_id()
    return "123456789"
end
```

6. Should display a message showing the id of the student when the method start is called



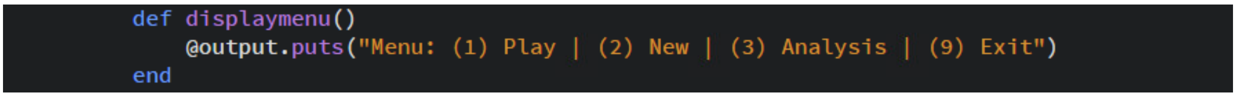
```
def start()
    @output.puts("Welcome to Deal or No Deal!")
    @output.puts("Designed by: " + self.created_by())
    @output.puts("StudentID: " + self.student_id())
end
```

7. Should display a starting message when the method start called



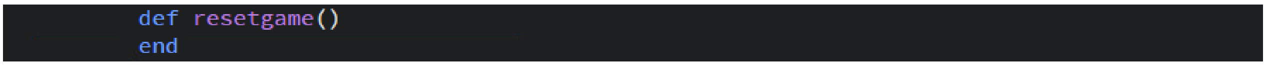
```
def start()
    @output.puts("Welcome to Deal or No Deal!")
    @output.puts("Designed by: " + self.created_by())
    @output.puts("StudentID: " + self.student_id())
    @output.puts("Starting game...")
end
```

8. Should display menu when method displaymenu called



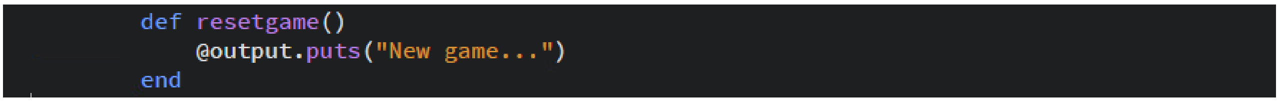
```
def displaymenu()
    @output.puts("Menu: (1) Play | (2) New | (3) Analysis | (9) Exit")
end
```

9. Should create a method called resetgame



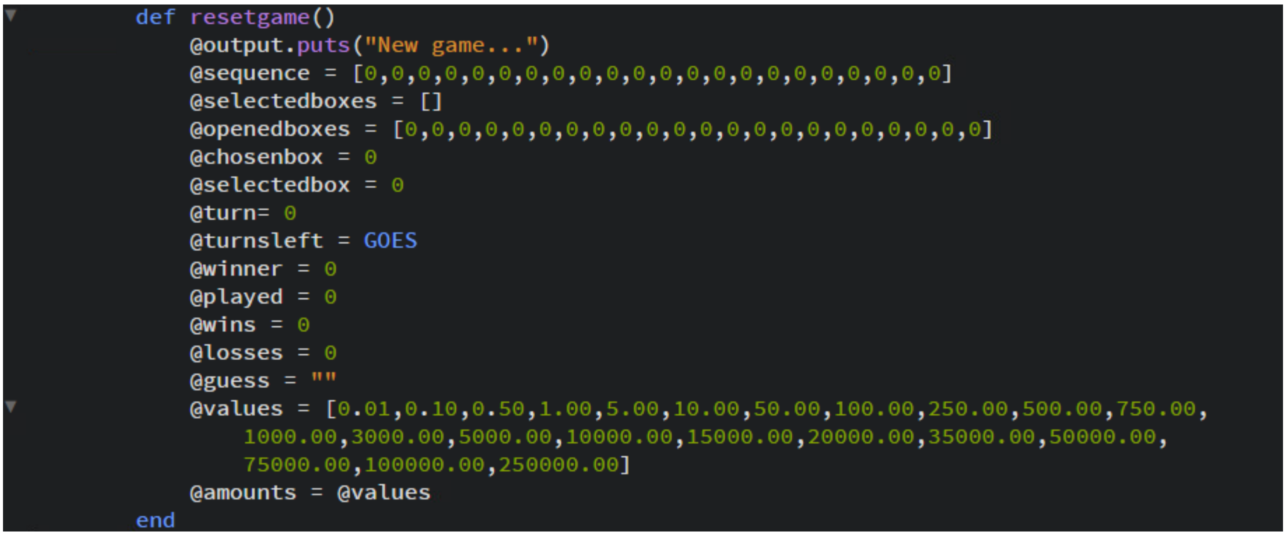
```
def resetgame()
end
```

10. Should display new game message when method resetgame called



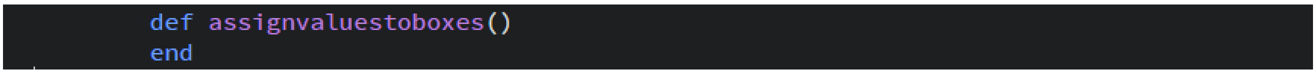
```
def resetgame()
    @output.puts("New game...")
end
```

11. Should set object variables to correct value when resetgame method called



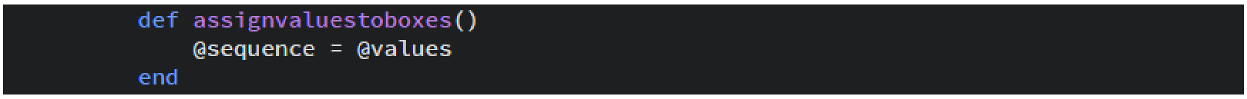
```
def resetgame()
    @output.puts("New game...")
    @sequence = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
    @selectedboxes = []
    @openedboxes = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
    @chosenbox = 0
    @selectedbox = 0
    @turn= 0
    @turnsleft = GOES
    @winner = 0
    @played = 0
    @wins = 0
    @losses = 0
    @guess = ""
    @values = [0.01,0.10,0.50,1.00,5.00,10.00,50.00,100.00,250.00,500.00,750.00,
        1000.00,3000.00,5000.00,10000.00,15000.00,20000.00,35000.00,50000.00,
        75000.00,100000.00,250000.00]
    @amounts = @values
end
```

12. Should create a method called assignvaluestoboxes



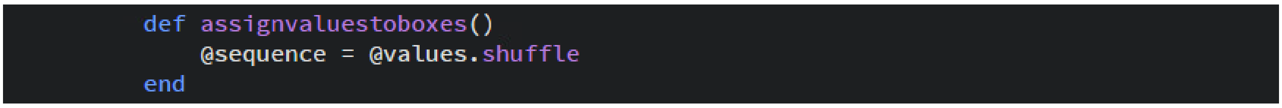
```
def assignvaluestoboxes()
end
```

13. Should provide 22 boxes (represented by @sequence[ ] array) containing amounts between 0.01 and 250000.00) when method assignvaluestoboxes called



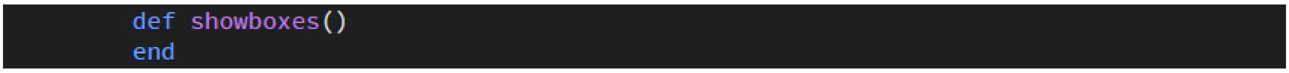
```
def assignvaluestoboxes()
    @sequence = @values
end
```

14. Should place each predefined amount (contained in the @values[] array) randomly into one empty box (represented by the @sequence[] array) when method assignvaluestoboxes called



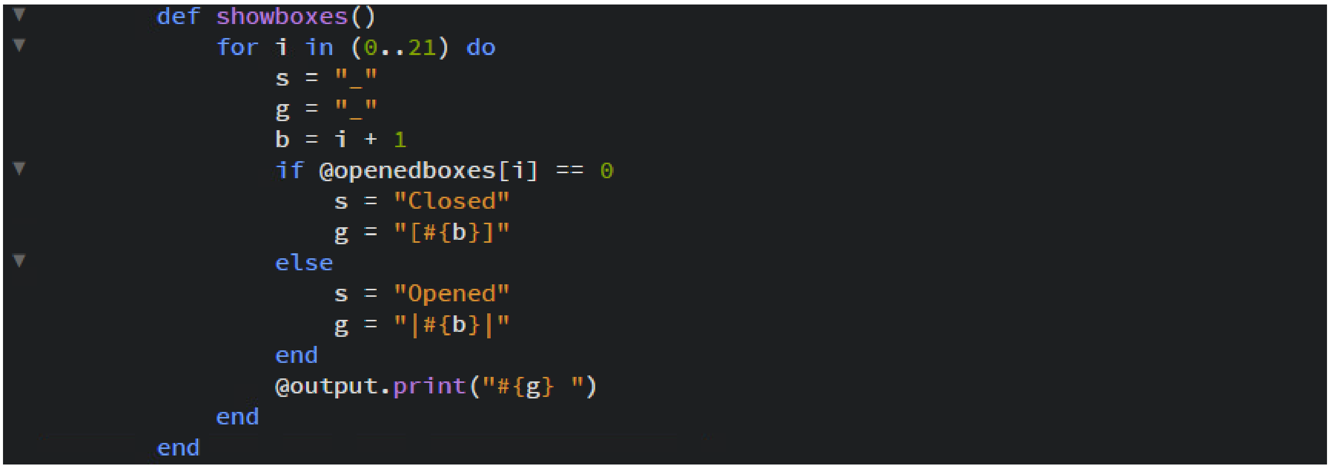
```
def assignvaluestoboxes()
    @sequence = @values.shuffle
end
```

15. Should create a method showboxes



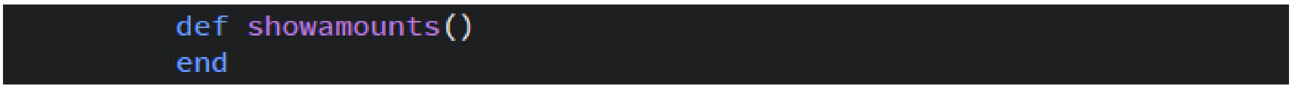
```
def showboxes()
end
```

16. Should list each box and its status (|opened|/[closed]) when showboxes method called



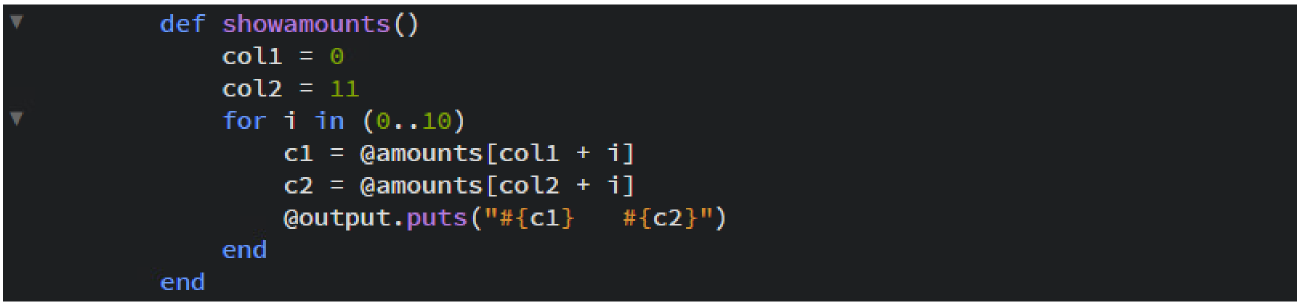
```
def showboxes()
    for i in (0..21) do
        s = "_"
        g = "_"
        b = i + 1
        if @openedboxes[i] == 0
            s = "Closed"
            g = "[#{b}]"
        else
            s = "Opened"
            g = "|#{b}|"
        end
        @output.print("#{g} ")
    end
end
```

17. Should create a method showamounts



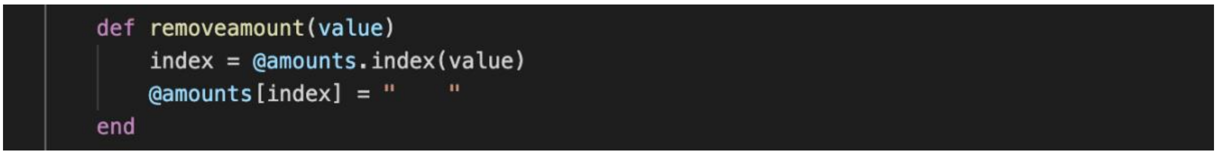
```
def showamounts()
end
```

18. Should show amounts in ascending order as two columns when method showamounts called



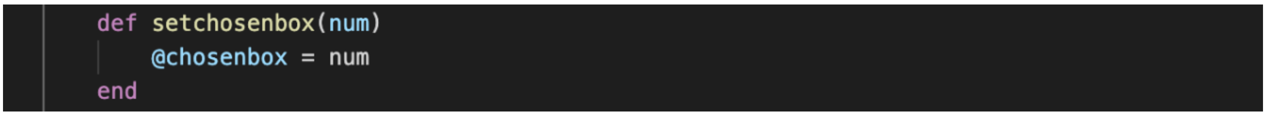
```
def showamounts()
    col1 = 0
    col2 = 11
    for i in (0..10)
        c1 = @amounts[col1 + i]
        c2 = @amounts[col2 + i]
        @output.puts("#{c1}    #{c2}")
    end
end
```

19. Should create a method removeamount that accepts parameter value



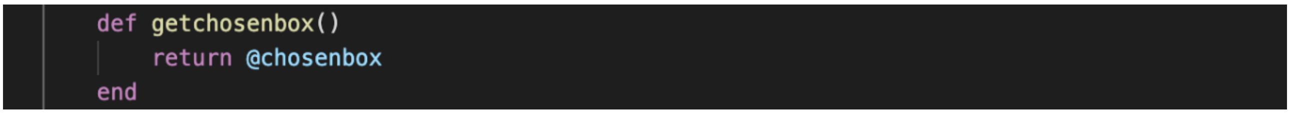
```
def removeamount(value)
    index = @amounts.index(value)
    @amounts[index] = "    "
end
```

20. Should create a method called setchosenbox that receives and stores a box number (contained in @chosenbox)



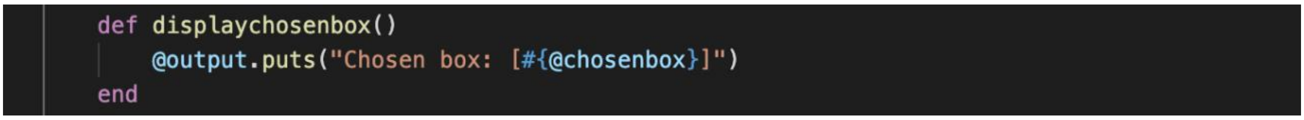
```
def setchosenbox(num)
    @chosenbox = num
end
```

21. Should create a method called getchosenbox that returns the box number (contained in @chosenbox)



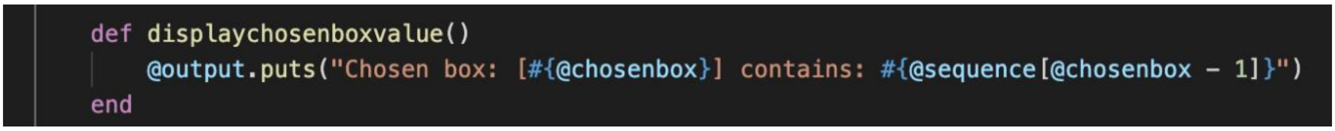
```
def getchosenbox()
    return @chosenbox
end
```

22. Should display a message representing the chosen box when method displaychosenbox called



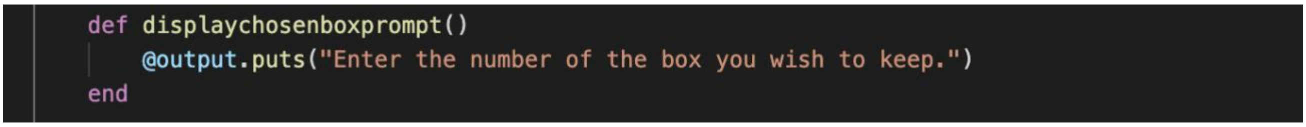
```
def displaychosenbox()
    @output.puts("Chosen box: [#{@chosenbox}]")
end
```

23. Should display a message containing the value stored in a chosen box when method displaychosenboxvlaue called



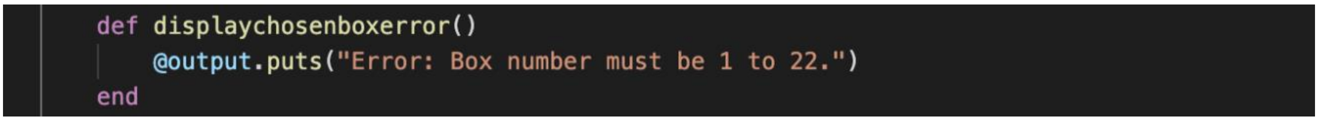
```
def displaychosenboxvalue()
    @output.puts("Chosen box: [#{@chosenbox}] contains: #{@sequence[@chosenbox - 1]}")
end
```

24. Should display a prompt requesting the user choose a box to keep until the end when method displaychosenboxprompt called



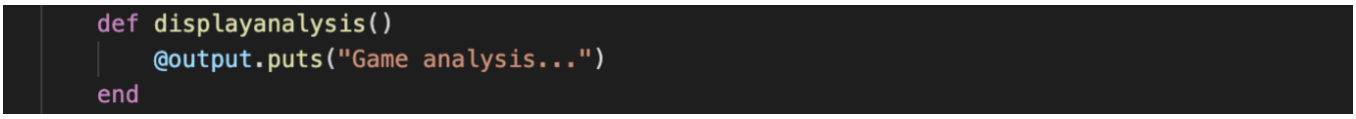
```
def displaychosenboxprompt()
    @output.puts("Enter the number of the box you wish to keep.")
end
```

25. Should display chosen box error when method displaycosenboxerror called



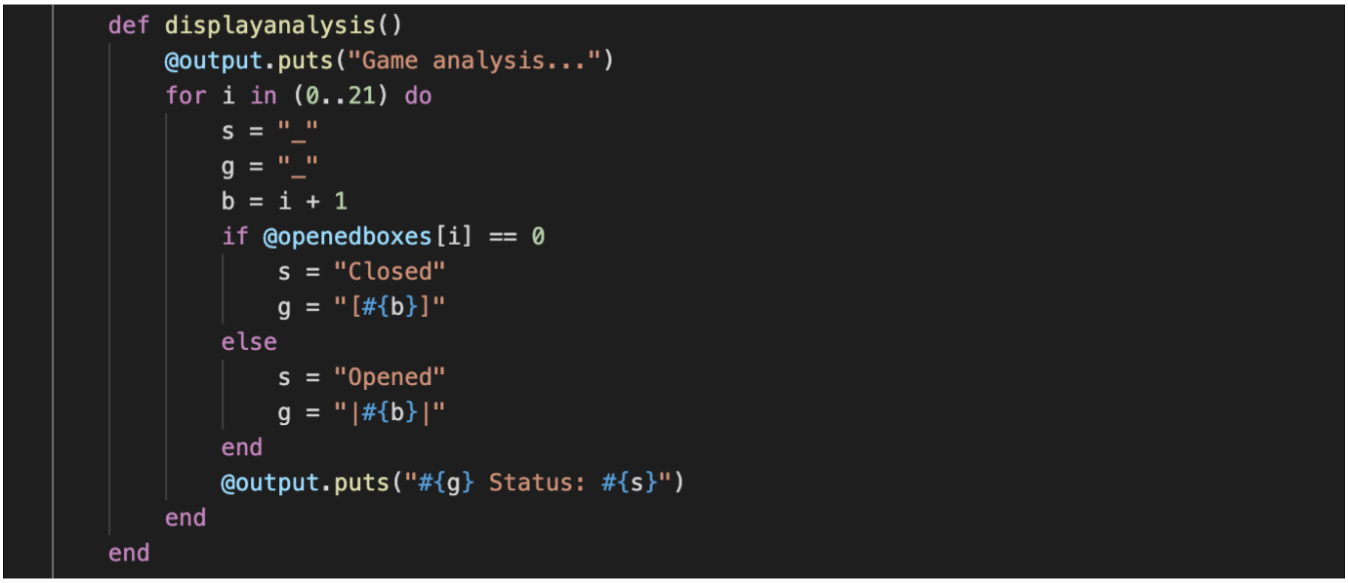
```
def displaychosenboxerror()
    @output.puts("Error: Box number must be 1 to 22.")
end
```

26. should display an analysis message within method displayanalysis



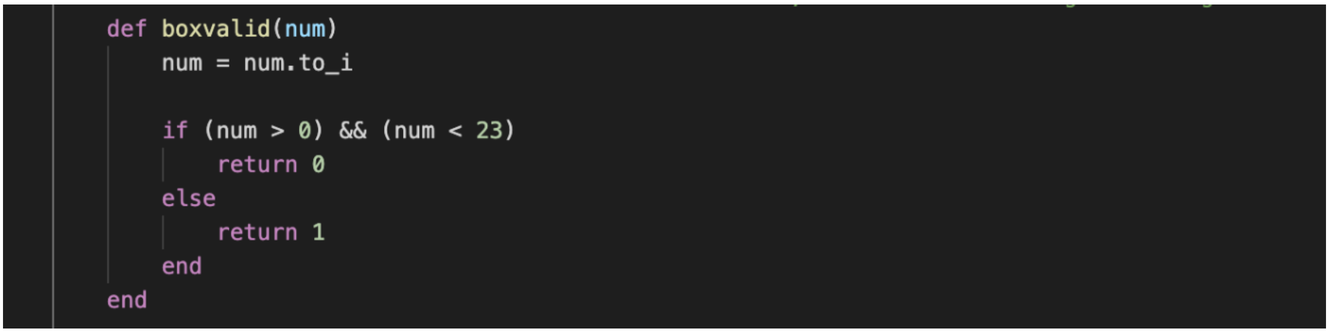
```
def displayanalysis()
    @output.puts("Game analysis...")
end
```

27. Should display a message and show the status (opened or cloased) of each box when method displayanalysis called



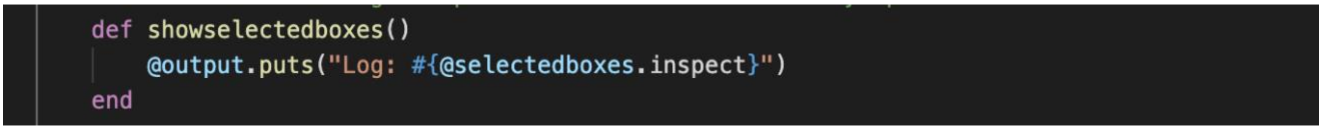
```
def displayanalysis()
    @output.puts("Game analysis...")
    for i in (0..21) do
        s = "_"
        g = "_"
        b = i + 1
        if @openedboxes[i] == 0
            s = "Closed"
            g = "[#{b}]"
        else
            s = "Opened"
            g = "|#{b}|"
        end
        @output.puts("#{g} Status: #{s}")
    end
end
```

28. Should check that entered box is between 1..22 when guess recieved by method boxvalid



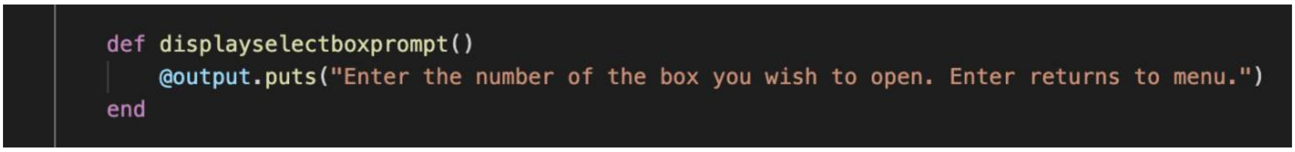
```ruby
def boxvalid(num)
    num = num.to_i

    if (num > 0) && (num < 23)
        return 0
    else
        return 1
    end
end
```

29. Should display log of boxes selected when method showboxesselected called



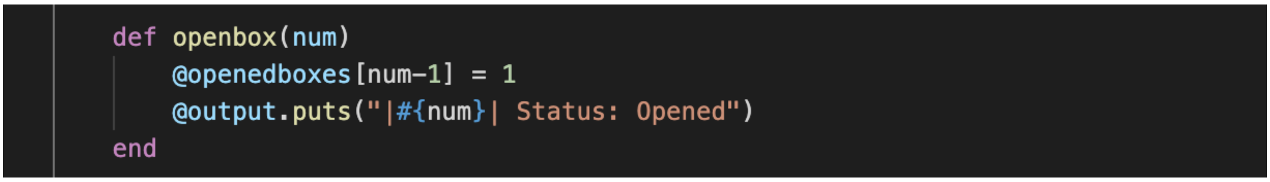
```ruby
def showselectedboxes()
    @output.puts("Log: #{@selectedboxes.inspect}")
end
```

30. Should display message requesting user to select box to open when method displayselectboxprompt called



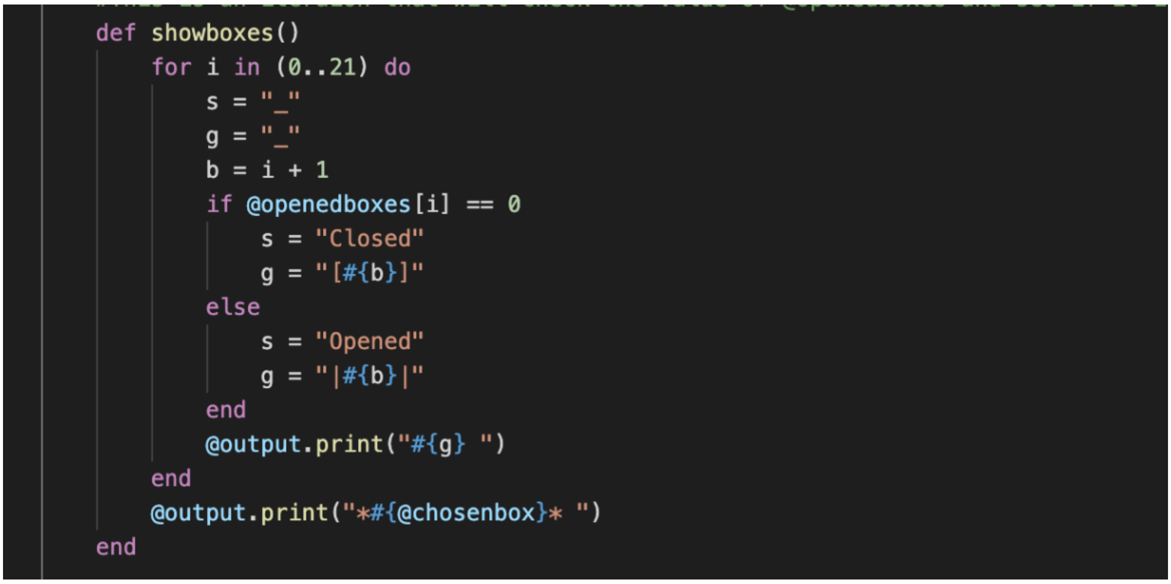
```ruby
def displayselectboxprompt()
    @output.puts("Enter the number of the box you wish to open. Enter returns to menu.")
end
```

31. Should display the status of a box (as opened) when method openbox recieves its associated box number



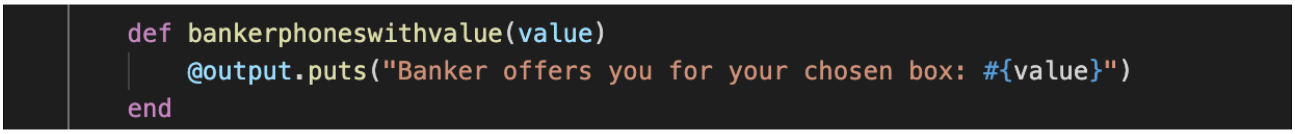
```ruby
def openbox(num)
    @openedboxes[num-1] = 1
    @output.puts("|#{num}| Status: Opened")
end
```

32. Should show chosen box when method showboxes called



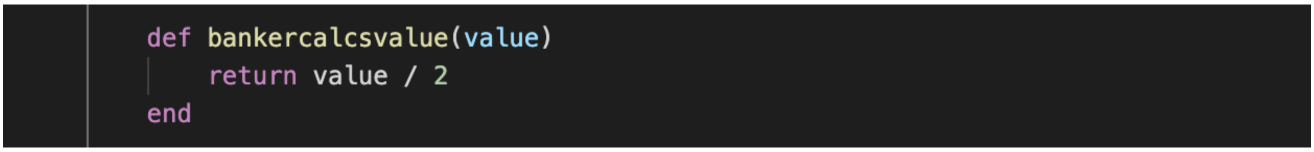
```
def showboxes()
    for i in (0..21) do
        s = "_"
        g = "_"
        b = i + 1
        if @openedboxes[i] == 0
            s = "Closed"
            g = "[#{b}]"
        else
            s = "Opened"
            g = "|#{b}|"
        end
        @output.print("#{g} ")
    end
    @output.print("*#{@chosenbox}* ")
end
```

33. Should display offer from banker when value received by method bankerphoneswithvalue



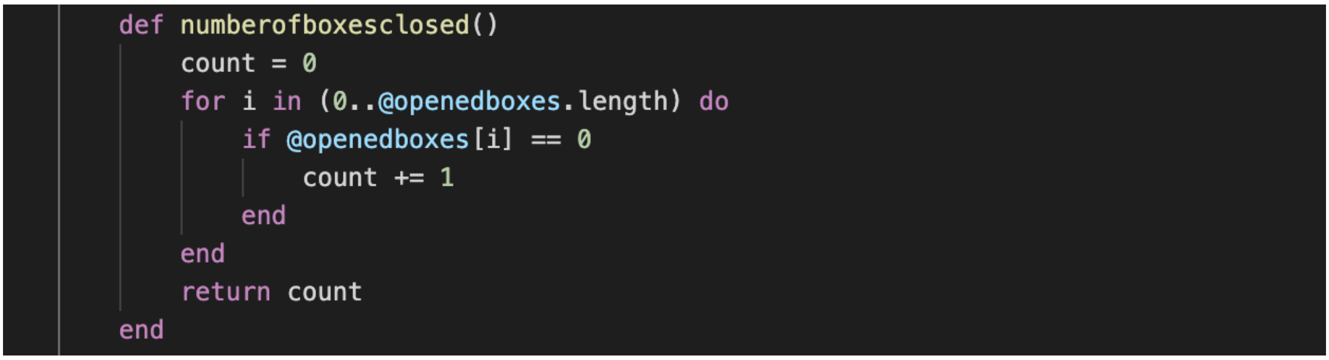
```
def bankerphoneswithvalue(value)
    @output.puts("Banker offers you for your chosen box: #{value}")
end
```

34. Should calculate and return offer from banker when value in box received by method bankercalcsvalue



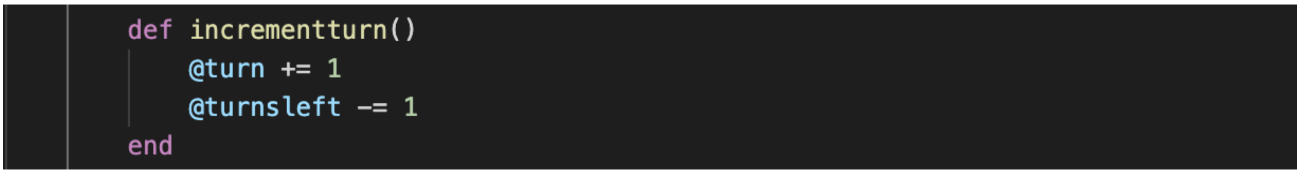
```
def bankercalcsvalue(value)
    return value / 2
end
```

35. Should return the number of boxes still closed when method numberofboxesclosed called



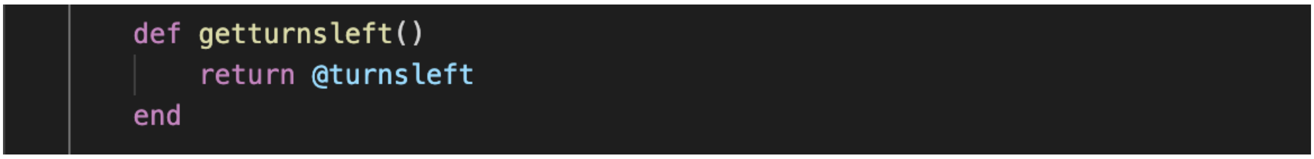
```
def numberofboxesclosed()
    count = 0
    for i in (0..@openedboxes.length) do
        if @openedboxes[i] == 0
            count += 1
        end
    end
    return count
end
```

36. Should create a method incrementturn which increases @turn by 1



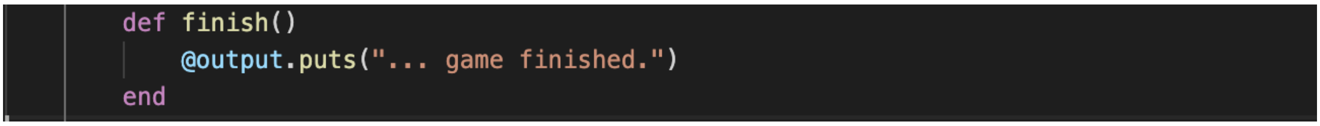
```
def incrementturn()
    @turn += 1
    @turnsleft -= 1
end
```

37. Should create a method getturnsleft which returns @turnsleft containing goes left



```
def getturnsleft()
    return @turnsleft
end
```

38. Should display an exit message when method finish called



```
def finish()
    @output.puts("... game finished.")
end
```