

GEDİZUNIVERSITY

Computer Engineering Department

COM401 (Fall-2014)
Software Engineering
Laboratory

HOMEWORK2 –

SAMPLE SOLUTION :

1- In the question, we are expected to write a method for each task.

a) It checks an e-mail address given as a parameter whether it is valid or not

```
def valid_email?(email_address)
  # Check the number of '@' signs
  if email_address.count("@") != 1 then
    return "Invalid e-mail address+here"

  elsif email_address =~ /^[a-z][a-z0-9+\-\.]+@[a-z].*(.com|.org|.net)$/i
  then
    return "Valid e-mail address"
  else
    return "Invalid e-mail address"
  end
end

puts valid_email?("xgamer123@gmail.com")
puts valid_email?("123gamer@gmail.com")
```

b) It extracts the dates written in Month Day, Year format like January 1, 1923 from a String below:

“Users of NYTimes.com Passes: — 1923–1980: Pass users have unlimited access to the New York Times archived articles outside the 1923–1980 date range. Each day of their pass, users may access up to five free articles published between the years 1923 through 1980.
— Pre-1923 and post-1980: Articles published before January 1, 1923 or after December 31, 1980 are free with your Pass and are not limited in any way.”

```
=begin
*****
*****
  b) A method that extracts the dates written in Month Day, Year format
  like January 1, 1923 from a String
*****
*****
=end
```

```
def matchDates(text)

  text.scan(/[a-z]{4,9}\s+\d{1,2},\s+\d{4}/i)
  # The September has the longest name among the months, and its length
  is 9
  #June has the shortest name among them, and its length is 4

end
```

```
str="Users of NYTimes.com Passes: — 1923–1980: Pass users have unlimited
access to the New York Times archived articles outside the 1923–1980 date
range. Each day of their pass, users may access up to five free articles
published between the years 1923 through 1980.
```

– Pre-1923 and post-1980: Articles published before January 1, 1923 or after December 31, 1980 are free with your Pass and are not limited in any way."

```
puts matchDates(str)
```

c) It checks an ISBN number as given as a parameter whether it is valid or not.

In here, there are many format for ISBN. That's why, there can be many solution for this question separately, or one solution containing each format. Just for the sake of simplicity we assume that we are looking for a regex only for ISBN dddddddd-d format.

```
=begin
*****
*****
  c) A method that checks an ISBN number (ISBN dddddddd-d format) as given
  as a parameter whether it is valid or not
*****
*****
=end
def check_isbn(isbn)
  if(isbn =~ /^ISBN\s\d{9}-\d/)
    return isbn+" is valid "
  else
    return isbn+" is invalid."
  end
end

puts check_isbn("ISBN 388053101-3") #valid
puts check_isbn("ISBN 388053101-32") #invalid
puts check_isbn("ISBN 3880531013") #invalid
```

d) It checks Turkish Identity Number as given as a parameter whether it is valid or not.

Note that the ID number is comprised of 11 digits which do not contain personal information.

```
=begin
*****
*****
  d) A method that checks Turkish Identity Number as given as a parameter
  whether it is valid or not.
*****
*****
=end
def turkish_id(id)
  if (id.size!=11)
    return "#{id} is an invalid citizenship number"
  end

  if id =~ /^[^0]\d{10}/
    return "#{id} is valid"
  end
  return "#{id} is invalid"
end
```

end

```
#some demonstration
puts turkish_id('123456789102') #invalid
puts turkish_id('02334567891') #invalid
puts turkish_id('12345678910') #valid
```

e) It checks a username given as parameters according to the following rules:

- A username length must be between 6 and 14.
- A username contains any lowercase letter (a-z), uppercase letter (A-Z), number (0-9), an underscore, or a hyphen

```
=begin
*****
*****
  e) A method that checks checks a username given as parameters according
to the following rules:
    - A username length must be between 6 and 14.
    - A username contains any lowercase letter (a-z), uppercase letter
(A-Z), number (0-9), an underscore, or a hyphen.
*****
*****
=end
```

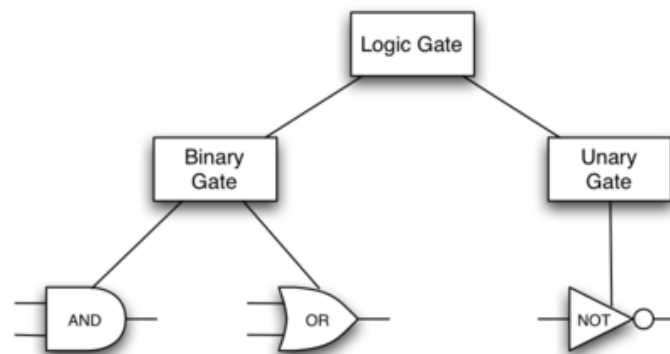
```
def check_username(username)
```

```
  if username =~ /[a-zA-z0-9_-]{6,14}/
    return "#{username} is valid"
  end
  return "#{username} is invalid"
```

end

```
puts check_username("guest")    #too short -invalid
puts check_username("theRulerofTheGame") #too long -invalid
puts check_username("guest?45")  # unknown symbol ? -invalid
puts check_username("guest12") #valid
```

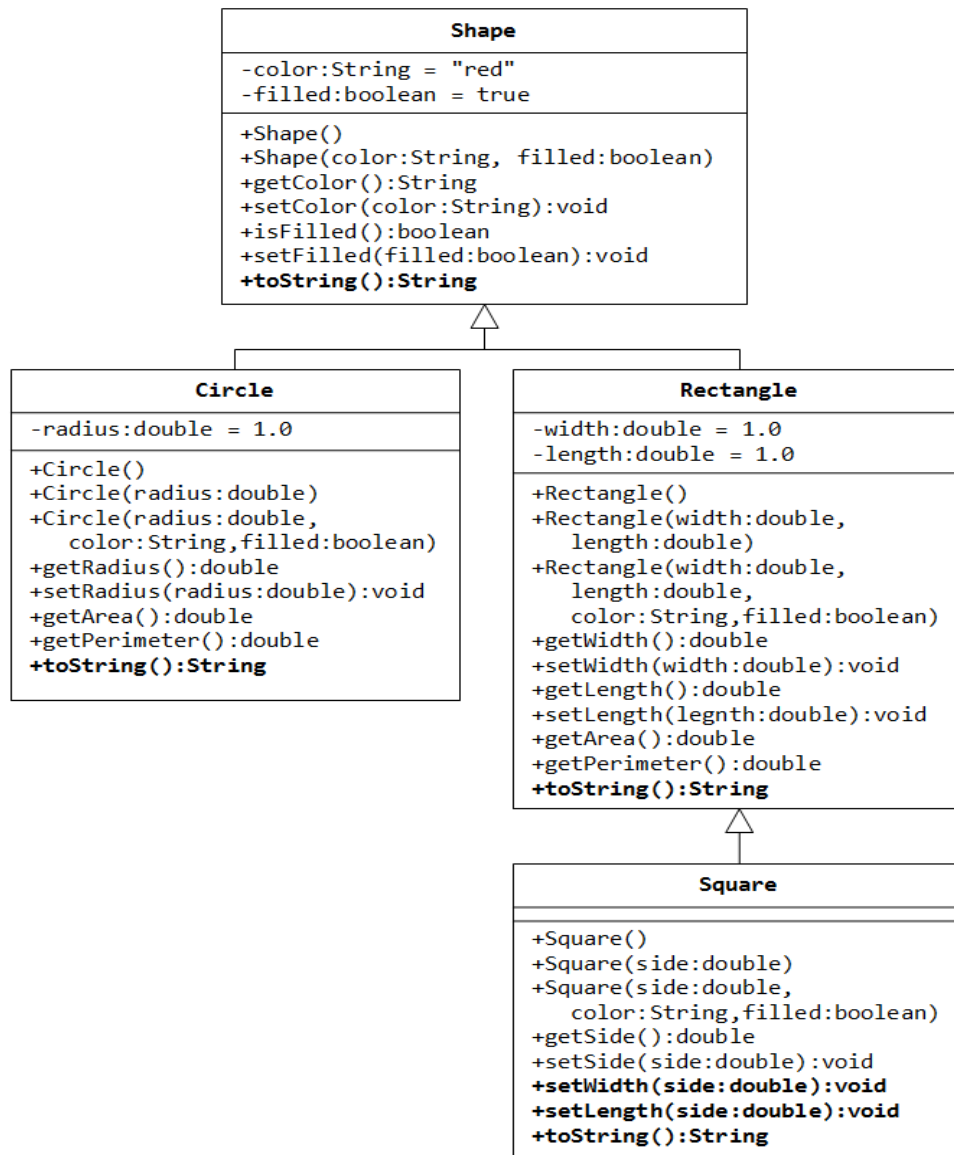
2- In this question, you are supposed to implement the figure below in Ruby.



You can find a sample source code for it on my website.

<http://arzumkaratas.weebly.com/sweng2014-2015.html>

3- In this question, you are supposed to implement the figure below in Ruby.



You can find a sample source code for it on my website.

<http://arzumkaratas.weebly.com/sweng2014-2015.html>

"On my honor I have neither given nor received unauthorized information regarding this work, and I am unaware of any violation of the Honor Code by others."