

Exercise : 1 Occurrence - Hash

Count the occurrences of various alphabet letters in an input string and store it in hash. Your ruby program should accept a string as an argument and display the hash as an output.

Exercise : 2 Classes and objects

Define a class Account with three attributes "name", "account_no" & "balance". Name and Balance should be set when creating an object of Account class, account_number should be auto increment. This Account class must have two methods, deposit() and withdraw().

Your program should take three arguments

1. customer one
2. customer two
3. transaction amount

Input Format:

customer_name1:account_balance1

customer_name2:account_balance2 transfer:transaction_amount

Exercise : 3 Array - Hash

Create a method for Array that returns a hash having 'key' as length of the element and value as an array of all the elements of that length.

Make use of Array#each.

Returned Hash should be sorted by key.

Your program should accept command line arguments.

Input argument: array-hash.rb ["abc','def',1234,234,'abcd','x','mnop',5,'zZzZ']"

output: {1=>["x", "5"], 3=>["abc", "def", "234"], 4=>["1234", "abcd", "mnop", "zZzZ"]}

Exercise : 4 Pascal - Yield

Print Pascal's triangle using 'yield'. Add exception handling for negative numbers

Eg: pascal(6) gives:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

Exercise : 5 Highlight Search Result

Write a search method that searches for a letter or word in a user input string and highlights the searched element. Also it should display the total no of searches found.

Eg: If user enters string to be searched as "Can you can a can as a canner can can a can?" and word to be searched is 'can' then output should be:

(Can) you (can) a (can) as a (can)ner (can) (can) a (can)?

Exercise : 6 Name - Raise

Create a class 'Name' with two attributes firstname and lastname. Neither of them can be blank and the first letter of firstname must be capital. Implement using 'raise'.

Try using custom Exceptions.

Exercise : 6 CSV

Read a CSV file:

Name, Empld, Designation

Jack, 15, Developer

Mary, 13, Designer

John, 12, Developer

Jane, 17, Designer

Johny, 19, Tester

Save into another file in the format:

Designers

Mary (Empld: 13)

Jane (Empld: 17)

Developers

John (Empld: 12)

Jack (Empld: 15)

Tester

Johny (Empld: 19)

1. Listing should be in ascending order of **Designation**.
2. Designation should be plural if it has more than one Employee.

Exercise : 7 Reverse Iterate

Implement a reverse iteration function for Array such that I should be able to do this

```
[2,4,6,8].reverse_iterate { |i| print "#{i} "}
```

It would print 8 6 4 2

note: Do not use any existing iterator for this.

Exercise : 8- Create Class from command line

1. Define a class dynamically. Class name should be taken from user by standard input(command line)
2. Then prompt user for a method name and a single line of code. This method should be defined as instance method in the class above dynamically with the code entered by user.
3. Tell user that the class and method is defined.
4. Then call this instance method and display the result

Ex.

```
Please enter the class name: User
Please enter the method name you wish to define: greet
Please enter the method's code: "Welcome from
#{self.class} class. I am #{self}"
```

```
--- Result ---
Hello, Your class User with method greet is ready.
Calling: User.new.greet:
"Welcome from User class. I am <User#123456>"
```

Exercise : 9 - Shopping list

Write a simple DSL for creating a shopping list. We should be able to specify the item name and quantity..

Something like.

```
sl = ShoppingList.new
sl.items do
  add("Toothpaste",2)
  add("Computer",1)
```

Exercise : 10: Creating Classes from CSV

Read a csv format file and construct a new class with the name of the file dynamically. So if the csv is persons.csv, the ruby class should be person, if it's places.csv, the ruby class should be places

Also create methods for reading and displaying each value in "csv" file and values in first row of csv file will act as name of the function.

Construct an array of objects and associate each object with the row of a csv file.

For example the content of the csv file could be

```
name,age,city  
gaurav,23,karnal  
vilok,23,hissar
```