Levi George

01/25/2021

CS 350: Programming Language Design

Prof. Amal Khalifa

Homework Assignment 1: Ruby Programming Language

A. When was Ruby Developed?
    a. **Ruby was developed in the 1990s**
B. What languages influenced the design of Ruby?
    a. **Ruby was influenced by Perl, Smalltalk, Eiffel, Ada, BASIC, and Lisp**
C. Discuss at least five features of Ruby as a programming language
    a. **Objected-Oriented design, default arguments, dynamic typing, exception handling, operator overloading are 5 program language level features**
D. Categorize the paradigm of Ruby, why have you chosen this paradigm?
    a. **Object-oriented, Imperative, Functional**
    b. Object-oriented programming languages include objects, since they are a feature of Ruby, it must be Object-oriented.
    c. Ruby is functional, this is because object-oriented languages offer a balance between operation and data. Operations include functions; thus, Ruby must be a functional language.
    d. Ruby is imperative because it each object therein needs to be told when to provide resources such as data or functions. It must be told how to trace through these data structures and what to get from them.
E. Search online for a sample program in Ruby, carefully explore the source code then answer the following:
    a. Describe variable declarations
        i. **Variable declarations are simple, you do not need to declare the type of your variable, simply it's name and value. (Dynamic typing). Additionally, arrays do not need to have all elements be the same type.**

ii. **Example declaration: (var = 10)**

b. Highlight and describe at least 3 “syntax” features

i. **Literals can be treated as objects, so you could use the function *.length* on the following string *“this is a string”* to get back the length**

ii. **You can use {} or “do … end” to write a block of code**

iii. **It supports variable interpolation, which allows you to quickly insert variables into strings without performing concatenation in the middle of the string. You simply write “I am a string that has been #{varName}” and the contents of varName will be placed in the string.**

c. Manually trace your program listing and give an output assuming a given input. What problem is this program trying to solve?

i. https://www.includehelp.com/ruby/reverse-a-string.aspx

Method (1): Without using library method

Copy

```
=begin
Ruby program to reverse a given string.
=end

puts "Enter the String:"
str1=gets.chomp

newstr= ' '

for  i in  1..str1.length
    newstr+=str1[str1.length - i]
end

puts "The reverse of #{str1} is #{newstr}"
```

ii.

iii. Method (1) is the method I dissected.

iv. This code takes user input using *puts* to output an input request message and then uses gets.chomp to retrieve user input.

v. Then it declares a new string to hold the final output

vi. For each letter in the user’s string, it will append that letter to the new string in reverse order by taking the letters at the end of the string and placing them at the beginning of the new string.

vii. **I suspect that this program is reversing the words that the user enters**

d. Give your opinion on the readability and writability of the language.

i. Ruby seems lightweight in terms of it's writing. You use indents to separate code logically but can use other means to isolate blocks of code. Additionally, dynamic typing allows you to forgo declaring data type. This makes it easier to type up than C++ or Java.
ii. While it is easier to type up compared to C++ and Java, I think this hurts its readability. It becomes cumbersome to sift through large swaths of code when you don't see clear separation between each large block. Being one who has programmed in strongly typed languages, Dynamic typing makes me physically uncomfortable.