

COMP 348: Principles of Programming Languages

Assignment 3 on C and Ruby

Summer 2020, sections AA and AB

May 30, 2020

Contents

1	General Information	2
2	Introduction	2
3	Ground rules	2
4	Your Assignment	2
4.1	The C Programming Language	3
4.2	Ruby	5
5	What to Submit	7
6	Grading Scheme	8

1 General Information

Date posted: Saturday May 30th, 2020.

Date due: Monday June 15st, 2020, by 23:59.

Weight: 9% of the overall grade.

2 Introduction

This assignment covers both Ruby and The C Programming Language, imperative as well as multi-paradigm programming.

3 Ground rules

You are allowed to work on a team of 3 students at most (including yourself). Each team should designate a leader who will submit the assignment electronically. **ONLY** one copy of the assignment is to be submitted.

This is an assessment exercise. You may not seek any assistance from others while expecting to receive credit. **(You must work strictly within your team)**. Failure to do so will result in penalties or no credit.

4 Your Assignment

This assignment consists of two parts:

1. The C Programming Language,
2. Ruby.

4.1 The C Programming Language

Q 1. Pointers

You know in Java, a called method from main can not change the value of an integer declared in main. For example:

```
public static void main() {  
    int a = 3;  
    CallFun( some parameters );  
    System.out.println(a);  
}
```

The above code will always print 3 on the screen no matter what happens in CallFun or its arguments.

Using pointers, rewrite the above program in C so that when CallFun is invoked the value of a is changed to 17 in the main function. You need to pass only one parameter to CallFun. You decide what that parameter is and what its type is.

Q 2. Pointers arithmetic & pointers to arrays and array elements

Use ONLY pointer arithmetic to implement the Selection Sort algorithm. You will also need to implement DisplayArray function that takes an array as a parameter and the number of elements it has and displays its content on the screen. The DisplayArray function should ONLY use pointer arithmetic.

For example:

```
int main() {  
    int arr[5] = {1, 13, 5, 17, 11};  
    DisplayArray (arr, 5);  
    SelectionSort (arr);  
    DisplayArray (arr, 5);  
}
```

Output:

1, 13, 5, 17, 11

1, 5, 11, 13, 17

Q 3. Memory management and records

Create a struct to represent a Book. The book has a title of type `char*` and a price of type float. The main program reads the number of books **n** to store. Then the program creates an array with **exactly n elements**.

The program then reads the information of each book and store them one by one in the array.

Develop a function **Display** that takes an array of Book and the number of books in the array. The Display function displays the content of the array on the screen.

Develop another function **AverageBookPrice** that takes an array of Book and the number of books in the array and returns the average book price of all books in the array.

Develop a function **Add** that reads the info of a Book from the user and:

1. Expands the size of the array by 1

Hint: create a new array and copy all books from old to new. Do not forget to free the old array

2. Adds the new book at the end of the newly created array

4.2 Ruby

This section consists of three parts that are related, but may be implemented independently.

Q 4. Classes in Ruby

Define the following class hierarchy in Ruby

1. Shape

- (a) : the class constructor receives no parameter.
- (b) : method `print()`: prints the name of the shape, perimeter, and area of the shape.
The name of the shape is the class name (i.e Shape for this class).
- (c) : method `perimeter()`: default method to be implemented by child classes; returning nil by default.
- (d) : method `area()`: default method to be implemented by child classes; returning nil by default.

2. Circle

- (a) : the class constructor receives the radius as the only parameter.
- (b) : method `perimeter()`: overridden, returns the perimeter of the circle.
- (c) : method `area()`: overridden, returns the area of the circle.

3. Rectangle

- (a) : the class constructor receives the height and width of the rectangle.
- (b) : method `perimeter()`: overridden, returns the perimeter of the rectangle.
- (c) : method `area()`: overridden, returns the area of the rectangle.

4. Ellipse

- (a) : the class constructor receives the semi-major and semi-minor axes (a and b).
- (b) : method `perimeter()`: not to be implemented.

- (c) : method `area()`: overridden, returns the area of the ellipse ($A = \pi ab$).
- (d) : method `eccentricity()`: additional method that returns the eccentricity of the ellipse ($e = \sqrt{a^2 - b^2}$).

Q 5. File / Text Processing

Write a Ruby program that reads text file that contains the shape information (see previous question). Every line in the text file consist of shape name and parameters to construct the shape. The program create every shape and calls the print method and displays the result on the screen. In case of errors (i.e. having negative values for sides, radius, etc.), and error message may be displayed. An example given below:

Input:

```
shape
rectangle 10 20
rectangle 0 10
circle 2
ellipse 2 4
ellipse -1 4
```

Output:

```
Shape, perimeter: undefined, area: undefined
Rectangle, perimeter: 60, area: 200
Rectangle, perimeter: 0, area: 0
Circle, perimeter: 12.5663706144, area: 12.5663706144
Ellipse, perimeter: undefined, area: 25.1327412287
Error: Invalid Ellipse
```

Q 6. Arrays and Hash

Extend the above program to display a statistics of the shapes after the text file is processes.

An example is given in the following

Output:

```
Statistics:
  Shape(s): 5
  Rectangle(s): 2
  Circle(s): 1
  Ellipse(s): 1
```

Use hashes to implement the above structure in memory.

Note that rectangles, circles, and ellipses are counted as shapes as well.

5 What to Submit

The assignment is to be submitted by the due date under the corresponding assignment box.

Your instructor will provide you with more details.

Submission Notes

Clearly include the names and student IDs of all members of the team in the submission.

Indicate the team leader.

IMPORTANT: You are allowed to work on a team of 3 students at most (including yourself).

Any teams of 4 or more students will result in 0 marks for all team members. If your work on a team, **ONLY** one copy of the assignment is to be submitted. You must make sure that you upload the assignment to the correct assignment box on Moodle. No email submissions are accepted. Assignments uploaded to the wrong system, wrong folder, or submitted via email will be discarded and no resubmission will be allowed. Make sure you can access Moodle prior to the submission deadline. The deadline will not be extended.

Naming convention for uploaded file: Create one zip file, containing all needed files for your assignment using the following naming convention. The zip file should be called a#_studids,

where # is the number of the assignment, and studids is the list of student ids of all team members, separated by (_). For example, for the first assignment, student 12345678 would submit a zip file named a1_12345678.zip. If you work on a team of two and your IDs are 12345678 and 34567890, you would submit a zip file named a1_12345678_34567890.zip. Submit your assignment electronically on Moodle based on the instruction given by your instructor as indicated above:

<https://moodle.concordia.ca>

Please see course outline for submission rules and format, as well as for the required demo of the assignment. A working copy of the code and a sample output should be submitted for the tasks that require them. A text file with answers to the different tasks should be provided. Put it all in a file layout as explained below, archive it with any archiving and compressing utility, such as WinZip, WinRAR, tar, gzip, bzip2, or others. You must keep a record of your submission confirmation. This is your proof of submission, which you may need should a submission problem arises.

6 Grading Scheme

- Q1 4 marks
- Q2 6 marks
- Q3 10 marks
- Q4 8 marks
- Q5 8 marks
- Q6 4 marks

Total: 40 marks.

References

1. GCC: <https://gcc.gnu.org/install/binaries.html>
2. Online C compiler: https://www.onlinegdb.com/online_c_compiler
3. C string manipulation: https://en.wikibooks.org/wiki/C_Programming/String_manipulation
4. <https://en.wikipedia.org/wiki/Ellipse>