Part 1: Short Questions

1. puts "Hello, World"

>Hello, World

1. Data Structures in Ruby
   1. Array – In Ruby, a single Array can hold items of mixed data types such as strings, integers and floats or even a method-call which returns some value.
   2. Hash – Hashes are another very useful, and widely used kind of thing that can be used to store other objects. Unlike arrays which are mere lists, Hashes are like dictionaries: You can look up a value from a Hash using a key.
   3. Set – Set implements a collection of unordered values with no duplicates. This is a hybrid of Array's intuitive inter-operation facilities and Hash's fast lookup. Unlike the other collection types, you must add a require statement to make use of the Set class. Also, unlike Array and Hash, Set does not have any kind of special literal syntax. However, you can pass an Array to Set#new. Alternatively, you can use Array#to\_set.
2. nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

nums.each do |num|

puts num

end

>1

>2

>3

>4

>5

>6

>7

>8

>9  
>10

1. personal\_info = { "first\_name" => "Eric",

"last\_name" => "Lee",

"tiger\_email\_id" => "szl0094",

"banner\_id" => 903750990

}

puts personal\_info["first\_name"]

puts personal\_info["last\_name"]

puts personal\_info["tiger\_email\_id"]

puts personal\_info["banner\_id"]

>Eric

>Lee

>szl0094

>903750990

1. A class is a blueprint or template or set of instructions to build a specific type of object. Every object is built from a class. Each class should be designed and programmed to accomplish one, and only one, thing. On the other hand, objects are the fundamental building blocks of applications from an object-oriented perspective. In programming terms, an object is a self-contained component that contains properties and methods needed to make a certain type of data useful. An object’s properties are what it knows, and its methods are what it can do.

Class Student

def initialize(age, name, major)

@stu\_age = age

@stu\_name = name

@stu\_major = major

end

end

and you can create object as follow –

stu1 = Student.new(“21”, “Eric”, “Computer Science”

1. Inheritance is a relation between two classes. It is like the relationship between cats and mammals, and mammals and animals. Encapsulation means that the internal representation of an object is hidden from the outside. Only the object can interact with its internal data. Public methods can be created to open a defined way to access the logic inside an object. Abstraction is a technique for arranging complexity of computer systems. It works by establishing a level of complexity on which a person interacts with the system, suppressing the most complex details below the current level.
2. Class Animal {

attr\_reader :name, :age

def initialize(name)

@name = name

end

def age=(a)

@age = a

end

}

1. Those are basically used for the initial setup of classes. The attr\_reader is short for the getter part of the class, attr\_writer is the setter part, and attr\_accessor is a combination of both. Using attr\_accessor will let you make changes to your instance variables as well as read them or see the values within.

Part 4 Thinking Assignment

Like the “sliding window” algorithm, scan the array from the left with index **i** and **j** which is **i + 1**.

If it’s not a fluctuation point, **i** should always smaller than **j. I**f at some point, **j** is smaller than **i**, then we know that we have a fluctuation point at index number **j**.