1.

a. Why does the update method in the Observer interface include a reference to the object being observed? Doesn't the observer know what object it is observing?

```
class Observer
        def update(o, arg)
                ...
        end
end
```

b. Describe one weakness of the following patterns
   i. Template Method

   ii. Strategy

   iii. Observer

c. Give a design example where you prefer abstract class over an interface?

d. Compare the *Template Method* and *Strategy* patterns by explaining one similarity and one difference?

e. How does Ruby change the way we implement interfaces?

2. You are tasked with developing the framework for a Graphics Engine Library that controls the game window creation, placement on the screen, resizing, and then removal. You should also allow an option for special window effects if the developer chooses to use it. Design the class structure and method(s) that allows this library to be used by several app developers. Clearly state any assumptions you make in your Class design as well as any design patterns used in the design.

Define what methods should be abstract and which ones would be implemented

3. You are a web app and software engineer for a new TV cable company that serves up channels a la carte. The software allows users to go online, and update their channel subscriptions and immediately receive the new channel on their TV. Likewise, they can unsubscribe, and immediately lose access to the channel. If the customers subscribe to a channel, they must wait at least 24 hours before removing the subscription. Design the class structure for this system. Clearly state any assumptions you make in your class design as well as any design patterns used in the design.

4. You are head of a new startup that sells a machine that can sort stuffed animals into 3 distinct groups. The machine has the ability to organize the stuffed animals by color, size, shape, or fluffiness based on the setting the user chooses. Design the software that the machine will use to sort the stuffed animals. Clearly state any assumptions you make in your class design as well as any design patterns used in the design.

5. Write a driver script that contains a **lambda** function that asks the user which two numbers she would like to add together and returns the result.