**105 points**

You are part of a team working on an application that provides the functionality to manage many sweepstakes. Marketing firms are likely purchasers and users of the backend application, and they can choose to use your functionality however they like! It is up to you, the developer, to create the backend application to implement the logic for the functions and pieces of functionality requested below.

**(5 points)** As a developer, I want consistent commits and descriptive commit messages.

**(5 points)** As a developer, I want to create a Contestant class that has a first name, last name, email address, and registration number.

**(10 points)** As a developer, I want to create a user interface for any information the application would need to get from the user. One example would be the functionality to assign a Contestant object a first name, last name, email address, and registration number.

**(15 points)** As a developer, I want to create a Sweepstakes class that uses the Dictionary data structure as an underlying structure. The Sweepstakes class will have the following methods with full implementation (write the functionality) of each method:

* Sweepstakes(string name)
* void RegisterContestant(Contestant contestant)
* Contestant PickWinner()
* void PrintContestantInfo(Contestant contestant)

**(10 points)** As a developer, I want to write an ISweepstakesManager interface with the following methods for a sweepstakes management system:

* void InsertSweepstakes(Sweepstakes sweepstakes)
* Sweepstakes GetSweepstakes()

**(10 points)** As a developer, I want to create a SweepstakesStackManager class that uses the Stack data structure as an underlying structure.

**(10 points)** As a developer, I want to create a SweepstakesQueueManager class that uses the Queue data structure as an underlying structure.

**(10 points)** As a developer, I want my SweepstakesStackManager class and SweepstakesQueueManager class to inherit from the ISweepstakesManager interface and implement the methods from the ISweepstakesManager interface using Stack and Queue methods.

**(5 points)** As a developer, I want to create a MarketingFirm class with functionality to create a sweepstakes.

**(10 points)** As a developer, I want to implement dependency injection in my MarketingFirm class so that I can utilize a sweepstakes manager.

**(5 points)** As a developer, I want to document (text file, canvas comment, etc.) where I am using dependency injection and what benefits it adds to the structure of my application.

**(10 points)** As a developer, I want to use the factory design pattern to allow a user to choose between a SweepstakesStackManager or a SweepstakesQueueManager to manage the sweepstakes objects.

**Bonus Points:**

**(5 points)** As a developer, I want to use the observer design pattern to notify all users of the winning contestant, with the winner of the sweepstakes getting a different message specifically congratulating them on being the winner.

**(5 points)** As a developer, I want to send an actual email to a sweepstakes winner using an MailKit API https://github.com/jstedfast/MailKit

Steps 1)

Create a marketing firm

Marketing Firm uses SweepstakeStackOrQueueManager

SweepstakesManager gets sweepstakes

To Start Simulation

1. Create Marketing Firm in Simulation class
2. Create Sweepstakes in MarketingFirm class
   1. Daily Sweepstakes
   2. Weekly Sweepstakes
   3. Monthly Sweepstakes
   4. Ongoing Sweepstakes
3. Add contestants in Sweepstakes class
4. Store on SweepstakesManager
   1. Stack Manager: InsertSweepstakes, Get Sweepstakes - ISweeptakesManager
   2. Queue Manager: InsertSweepstakes, Get Sweepstakes - ISweeptakesManager
5. Select Sweepstakes in MarketingFirm class.
6. Select Winner in MarketingFirm class