Panthera Finance Senior Developer Exercises

To gain an understanding of your abilities you will be given 4 different exercises to complete offline.

Once submitted, your code will be reviewed by software developers with a focus on:

* Appropriate use of languages, libraries, patterns and methodologies to solve the problem
* Code that is easily understood and maintainable by other developers
* Code that is robust and handles any errors appropriately
* A solution that compiles and functions as intended.

When you have completed your exercises, please ZIP your solution and forward it along with any other instructions to run the solutions. Please return all completed exercises to [dmurrell@pantherafinance.com.au](mailto:dmurrell@pantherafinance.com.au)

It should not take more than a couple of hours to complete all 4 exercises.

If you do not have time to complete them all please just finish those that you feel are best suited to showcasing your skillsets and send through.

Please try to return within the next 2-3 days.

## **Exercise 1**

Write a program that:

* Prints the numbers from 1 to 100.
* For multiples of 5 print “Flip” instead of the number.
* For multiples of 7 print “Flop” instead of the number.
* For multiples of 5 and 7 print “Flip Flop” instead of the number.

You can use any technology that suits you best.

## **Exercise 2**

Using your choice of test frameworks, implement the code to make the following test method pass:

public void TestReceipt()

{

var receipt = new Receipt();

receipt.AddItem(1, "Newspaper", 1.50);

receipt.AddItem(1, "Milk", 3);

receipt.AddItem(2, "Bread", 2.50);

var expected =

@"1 Newspaper @ $1.50 = $1.50

1 Milk @ $3.00 = $3.00

2 Bread @ $2.50 = $5.00

Sub Total = $9.50

Tax (10%) = $0.95

Total = $10.45";

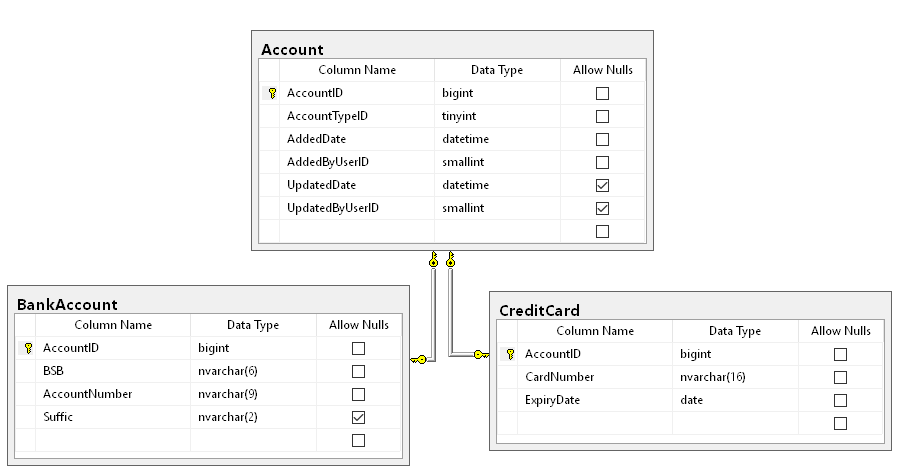
Assert.AreEqual(expected, receipt.ToString());

}

## **Exercise 3**

Given the SQL Server table structure below; Write a stored procedure that inserts a new credit card record and returns the new AccountID value.

**Important Note: Account.AccountID is an identity column with a seed value of 1**



## **Exercise 4**

Build a game review web page to allow gamers to list and review games. Each game listed will have the following properties:

* **Title** – The name of the game
* **Description** – Detailed description of the game
* **Rating** – The average rating based on votes from users (1- 5 stars)

For example:

|  |  |  |
| --- | --- | --- |
| World of Warcraft | Vivamus purus eros, aliquet malesuada gravida  at, fringilla vel elit. Mauris vestibulum, erat at  volutpat semper, metus enim faucibus nunc, in  ultrices magna enim in justo | ✰✰✰✰ |
| League of Legends | Integer magna magna, iaculis euismod  tincidunt a, cursus ac dolor. Aenean quis  egestas diam. Pellentesque purus ipsum,  elementum sit amet malesuada eget, aliquet  eu magna. Nullam magna massa, sodales nec  imperdiet quis, consectetur eget nisl. Aenean  eget velit in eros porta dictum. Sed eu dui | ✰✰ |
| Final Fantasy | Lorem ipsum dolor sit amet, consectetur  adipiscing elit | ✰✰✰ |

### User Story

1. As a gamer I want to edit the descriptions so that everyone can see the latest game information.

2. As a gamer I want to vote for my favourite games.

3. As a gamer I want to view games based on rating so that I can find the most popular ones.

### Design Considerations

* For features that aren’t required, like creating a new game, can be done programmatically.
* You can use any technology of your choice e.g:
  + ASP.NET MVC, Angular/Reactjs/Nodejs, PHP
  + SQL Server/XML or in-memory database
  + WCF/REST services, if any
  + Third Party Libraries like Boostrap,NHibernate, EntityFramework, etc
  + Any other tool that you consider useful for this task.
* UI design is a consideration, but we’ll mainly review the technologies used, and of course the code.