

<b>Student's Name: Andrew Hoon</b>			
<b>Overall Grade:</b>	<b>98.57%</b>		
<b>Assignment Name</b>	<b>Points Earned</b>	<b>Total Potential Points</b>	<b>Attendance</b>
Project 1 - FortuneTeller	100	100	
Project 2 - FortuneTellerMethods	90	100	<b>99.43%</b>
Project 3 - Virtual Pet	100	100	
Project 4 - Bank Account	100	100	
Project 5 - Tuscany Villa	100	100	
Project 6 - MVC Review Site	100	100	
Project 7 - Wardrobe MVC	100	100	

<b><u>TASK TO BE COMPLETED</u></b>	<b><u>POINTS EARNED</u></b>	<b><u>POSSIBLE POINTS</u></b>	<b><u>Student's Name</u></b>	andrew hoon	<b><u>Student's Email</u></b>
Geting User input (Part 1)	30	30			
Commutiing the user input (Part 2)	30	30			
Outputting the user input to the console (	30	30			
Initial Score	90	90			
Convert to 90 points	90	90			
			<b>Comments:Great Job!! I like that you allowed the</b>		
Overall Code Quality	10	10			
Late Submission			(enter negative value if late submission)		
<b>Total Score</b>	100	100			

<b><u>TASK TO BE COMPLETED</u></b>	<b><u>POINTS EARNED</u></b>	<b><u>POSSIBLE POINTS</u></b>	<b><u>Student's Name</u></b>
Part 1 - This is the same as your last FortuneTeller	5	5	Andrew Hoon
Create a Method that greets the user, takes two string parameters, returns a string.	5	5	
Create a Method that calculates the number of years before the user retires. If the user's age is an odd number, then they will retire in ____ years, or ____ years if their age is an even number. This method must have an 'int' as a parameter (such as the user's age) and must return an 'int' representing the number of years before retirement to the Main method.	5	5	

<p>Create a Method that determines the vacation home of the user. This method must take an 'int' as a parameter (such as the user's number of siblings) and return a 'string' of the vacation home's location.</p>	5	5	
<p>Create a Method that determines the user's future mode of transportation. This method should take a 'string' as a parameter (such as the user's favorite color) and return a 'string' of the mode of transportation. If the user enters anything other than a ROYGBIV color, their mode of transportation should be a squeaky shopping cart.</p>	5	5	

Create a Method that determines how much money the user will have when they retire. This method should take an 'int' as a parameter (such as birth month) and return a 'double' of their bank account amount.	5	5	
Create a Method that judges how good the fortune is. This method should have no parameters and be "return type void".	5	5	
Allow the user to quit the program at any point where the program is looking for user input.	5	5	
Allow the user to restart the program at any point where the program is looking for user input.	0	5	
Program should be able to handle whether or not a user inputs capital or lowercase letters.	5	5	
PART 3	5	5	
Initial Score	50	55	

Convert to 90 points	82	90	
			<b>Comment:</b>
Overall Code Quality	8	10	<p>This project was very well done. The amount of commenting that the coder used helped make the code significantly more readable. Usually, the Main method is above all of the other methods, and with some extraneous spacing, the code quality was given 8/10. I am particularly impressed by the Judgment() method! You made it interesting and unique! Great job overall!!</p>
Late Submission			
Total Score	90	100	

<b><u>TASK TO BE COMPLETED</u></b>	<b><u>POINTS EARNED</u></b>	<b><u>POSSIBLE POINTS</u></b>	<b><u>Student's Name</u></b>
Create a console application in Visual Studio called VirtualPet	2	2	Andrew Hoon
Create a repository called virtual-pet	2	2	
Create a README.md file explaining what this project will involve	2	2	
Created VirtualPet class	5	5	
Properties (at least three)	5	5	
Properties (at least three)	5	5	
Properties (at least three)	5	5	
Methods (at least three)	5	5	
Methods (at least three)	5	5	
Methods (at least three)	5	5	
Use your VirtualPet class and methods in the Main method of the Program class	5	5	
Initial Score	46	46	
Convert to 90 points	90	90	
			<b>Comment:</b>

Overall Code Quality	10	10	<p>This project meets all of the requirements of the assignment. The code is well organized and comments are used thoroughly throughout. The user experience is very fun. Game Save does work and is a fantastic use of StreamWriter! The code is complex and well thought out. Great job!</p>
Late Submission			
Total Score	100	100	



<b><u>TASK TO BE COMPLETED</u></b>	<b><u>POINTS EARNED</u></b>	<b><u>POSSIBLE POINTS</u></b>	<b><u>Student's Name</u></b>
<b>Menu</b>			Andrew Hoon
View Client Information	5	5	
View Account Balance - Checking	5	5	
View Account Balance - Savings	5	5	
View Account Balance - Reserve	5	5	
Deposit Funds	5	5	
Withdraw Funds	5	5	
Exit	5	5	
<b>Checking Account Class</b>			
Inherits Account base class	5	5	
Minimum 2 fields	5	5	
Minimum 2 properties	5	5	
Minimum 1 Constructor	5	5	
<b>Reserve Account Class</b>			
Inherits Account base class	5	5	
Minimum 2 fields	5	5	
Minimum 2 properties	5	5	
Minimum 1 Constructor	5	5	
<b>Savings Account Class</b>			
Inherits Account base class	5	5	
Minimum 2 fields	5	5	

Minimum 2 properties	5	5	
Minimum 1 Constructor	5	5	
<b>Objects</b>			
Minimum 1 object instantiated from the Checking Account Class	5	5	
Minimum 1 object instantiated from the Reserve Account Class	5	5	
Minimum 1 object instantiated from the Savings Account Class	5	5	
<b>Other Required Tasks</b>			
Use StreamWriter to create account summary files that track transactions for each account type (each type has its own text file). Files should be located in the Debug Folder and should include:	5	5	
Name of Client	1	1	
Client Account Number	1	1	
Account Type	1	1	
Each transaction should be its own line	1	1	

Each transaction should show date and time of transaction (must use DateTime Class)	1	1	
Each transaction should show a "+" for deposit and a "-" for withdrawal	1	1	
Each transaction should show the transaction amount	1	1	
Each transaction should show the new current balance after the transaction	1	1	
Initial Score	123	123	
Convert to 90 points	90	90	
			<b>Comment:</b>
Overall Code Quality	10	10	Wow! This is the most creative version of the Bank Account project I've seen to date! The code is beautifully organized and commented and uses all proper naming conventions. The console experience is fantastic. StreamWriter is used correctly. This project meets and exceeds all requirements of the assignment. Great work!!
Late Submission			
Total Score	100	100	

<b><u>TASK TO BE COMPLETED</u></b>	<b><u>POINTS EARNED</u></b>	<b><u>POSSIBLE POINTS</u></b>	<b><u>Student's Name</u></b>
Create the base HTML in a file called index.html	5	5	Andrew Hoon
Use semantic elements where appropriate	5	5	
Use existing bootstrap classes to develop the mobile version first	5	5	
Address xs, sm, md, and lg sizes as shown in the mockups	5	5	
Add a custom stylesheet to incorporate more features	5	5	
Initial Score	25	25	
Convert to 90 points	90	90	
			<b>Comment:</b>
Overall Code Quality	10	10	This project meets all of the requirements of the assignment. The code is easy to read and well organized. Adjusting for screen size is done very well. Great work!
Late Submission			
Total Score	100	100	

<b><u>TASK TO BE COMPLETED</u></b>	<b><u>POINTS EARNED</u></b>	<b><u>POSSIBLE POINTS</u></b>	<b><u>Student's Name</u></b>
<b>Menu</b>			Andrew Hoon
Create Review Model	5	5	
Create Category Model	5	5	
Review Model Has Appropriate Properties	5	5	
Index View Styling on Review Site	5	5	
Details View Styling on Review Site	5	5	
Bootstrap: Header	5	5	
Bootstrap: Navbar	5	5	
Bootstrap: Footer	5	5	
Bootstrap: Rows	5	5	
Bootstrap: Columns	5	5	
Documented throughout	5	5	
Initial Score	55	55	
Convert to 90 points	90	90	
			<b>Comment:</b>
			This project meets all of the requirements of the assignment. The code is well organized and documented. The styling is appropriately done and original. Nice work!
Overall Code Quality	10	10	
Late Submission			
Total Score	100	100	

<b><u>TASK TO BE COMPLETED</u></b>	<b><u>POINTS EARNED</u></b>	<b><u>POSSIBLE POINTS</u></b>	<b><u>Student's Name</u></b>
Project set up			Andrew Hoon
Create a project/repository called wardrobe	2.5	2.5	
Create a README file in your repository explaining what this project will involve	2.5	2.5	
Controllers for each model	5	5	
Views for each model	5	5	
Appropriate table organization of clothing, including type	5	5	
Customize the index view to properly show the properties of each item	5	5	
Customize the details view to properly show the properties of each item	5	5	
Use photos properly where appropriate	5	5	
/Home/Index.cshtml should be updated for your app	5	5	
<b>Bootstrap/CS S</b>			
Updated navbar	5	5	
Thumbnails	2	2	
At least 3 other Bootstrap components of your choice	3	3	

Responsive layout	5	5	
Initial Score	55	55	
Convert to 90 points	90	90	
			<b>Comment:</b>
			<p>This project does a fantastic job of meeting all of the requirements. The code is really well organized, commented and easy to read. The table organization is truly impressive. Next time you submit a database-first project, please make sure to include a SCRIPT of your database so that it's easier to run.</p>
Overall Code Quality	10	10	Awesome job!
Late Submission			
Total Score	100	100	