Grading Rubric — Homework 1

Criteria \ Points	Outstanding	Mostly Complete	Missing Some Important Parts	Needs a lot of work / Many parts missing	Missing/Not Included
	4	3	2	1	0
Used Git, has many commits, over time, wrote appropriate commit comments, set up Git correctly with git config					
Cloning repository results in a working site, no broken links, no absolute paths					
Multi-page, hyperlinked set of pages					
Uses Bootstrap and has single column and multi-column pages					
Uses a separate CSS file and all pages use it for a consistent look					
Has a navigation bar or menu that works, has a table, has lists					
Has GitHub Pages page as a Portfolio, has links to all work, has students name, course name/title					
Portfolio: all work is described, is complete, has good examples, is organized, professional and reflects well on the student; has syntax highlighted code, descriptions and link to assigned HTML page that displays in browser properly					
Totals					

Score	/ 32

Grading Rubric — Homework 2

Criteria \ Points	Outstanding 4	Mostly Complete 3	Missing Some Important Parts 2	Needs a lot of work / Many parts missing 1	Missing/Not Included 0
Uses Git feature branching correctly; used a new branch for most work					
Final work is merged back into the master branch					
Has a wireframe of initial design					
Uses Javascript effectively					
Mostly uses JQuery to access the DOM					
Application takes input from the user; doesn't fail if data is not provided correctly					
Button or other UI action causes Javascript to be executed in the current page; is responsive					
Uses JQuery to modify and add to the DOM to display the results					
Code is adequately and appropriately commented					
Code, screenshots, descriptions and the working page(s) are nicely arranged into the Portfolio; Portfolio is organized and easy to read; page can be demoed					
Totals					

Score	/ 40

Grading Rubric — Homework 3

Criteria \ Points	Outstanding 4	Mostly Complete 3	Missing Some Important Parts 2	Needs a lot of work / Many parts missing 1	Missing/Not Included 0
Uses Git branching correctly; used a new feature branch for most work					
Commits show that code was written from scratch and not copy and pasted then fixed					
Code compiles and is functionally equivalent to the Java version					
Built-in typenames like string and object are lowercase					
Method names and public fields start with upper case, private and local variables start with lower case					
Interface typename starts with an I, like IStack					
Node class uses properties					
Code is adequately and appropriately commented using XML comments					
Code and screenshots are nicely displayed in the Portfolio; Portfolio is organized and easy to read; page contains more than just the code and screenshots					
Totals					

Score	/ 36

Grading Rubric — Homework 4

Criteria \ Points	Outstanding 4	Mostly Complete 3	Missing Some Important Parts 2	Needs a lot of work / Many parts missing 1	Missing/Not Included 0
Uses Git branching correctly; used a new feature branch for each feature					
Task 2: Has main index page with links; uses @Html.ActionLink					
Task 3: Full cycle - form elements - GET request - Controller action method with no parameters - uses Request object - returns content to client					
Task 4: Same cycle but uses POST and FormCollection					
Task 5: Server side content (loan calculator or equivalent); form data, uses model binding in controller					
Task 5: Form data is either validated or doesn't cause an error if incorrect or missing					
Both C# and View code is appropriately commented					
Code and screenshots are nicely displayed in the Portfolio; Portfolio is organized and easy to read; page contains more than just the code and screenshots					
Totals					

Score	/ 32

Grading Rubric — Homework 5

Criteria \ Points	Outstanding	Mostly Complete	Missing Some Important Parts	Needs a lot of work / Many parts missing	Missing/Not Included
	4	3	2	1	0
Has a web-based version of the form and it works					
Has working scripts to create and drop the database and table					
Has correct model class and DbContext class (in DAL folder); model wasn't auto-generated from the db using EF					
Model class uses data annotations such as Display, Required or StringLength, form values are validated					
Web.config has correct connection string					
Uses GET-POST-Redirect pattern					
Uses a strongly typed view; controller action method uses a parameter of this type					
Code is appropriately commented					
Code and screenshots are nicely arranged into the Portfolio; Portfolio is organized and easy to read					
Can tell from the screenshots that it works as intended: new request and view all requests					
Totals					

Score	/ 40

Grading Rubric — Homework 6

Criteria \ Points	Outstanding	Mostly Complete	Missing Some Important Parts	Needs a lot of work / Many parts missing	Missing/Not Included
	4	3	2	1	0
Website works and clearly connects to the db and reads data from tables					
Has Browse Products feature					
Browse by product category and subcategory works; user can find a particular product					
User can successfully add a review to a product; review can be seen in the database					
Uses mostly strongly typed views					
Views look nice, not just CRUD auto-generated					
EXTRA: Shows images or thumbnail images for products					
Code is appropriately commented					
Code and screenshots are nicely arranged into the Portfolio; Portfolio is organized and easy to read					
Totals					

Score	/ 36