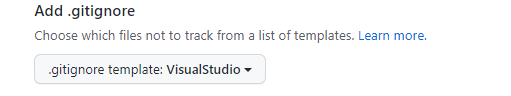
**94%**

**Complete the following test in 4 hours. You may use your textbook, labs and lecture notes.**

**All code submitted must be your original work. Code must be committed and pushed to your git repository with the solution name HOT1 to receive credit.**

****

**Add your instructor as a collaborator to the repository.**

**Implement all of the below exercises in one project.**

**HomeController and Layout file are optional for this test only.**

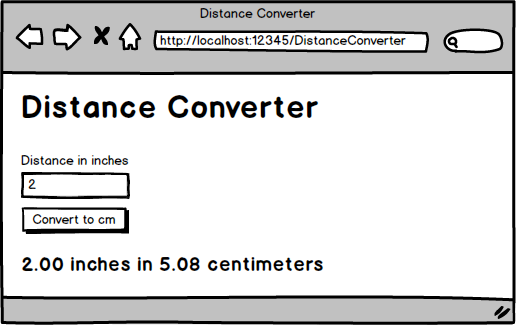
**DistanceConverter [50 pts]**

Write a program that converts distances between inches and centimeters.

* The program must contain a constant named **CM\_PER\_IN** which holds the number of inches per centimeter (2.54 cm per inch).
* Prompt the user for a distance in inches.
* When the user submits the form, display the distance in both inches and centimeters as follows:

“2.00 inches is 5.08 centimeters”

* **See rubric on next page for more details.**



**DistanceConverter - Rubric**

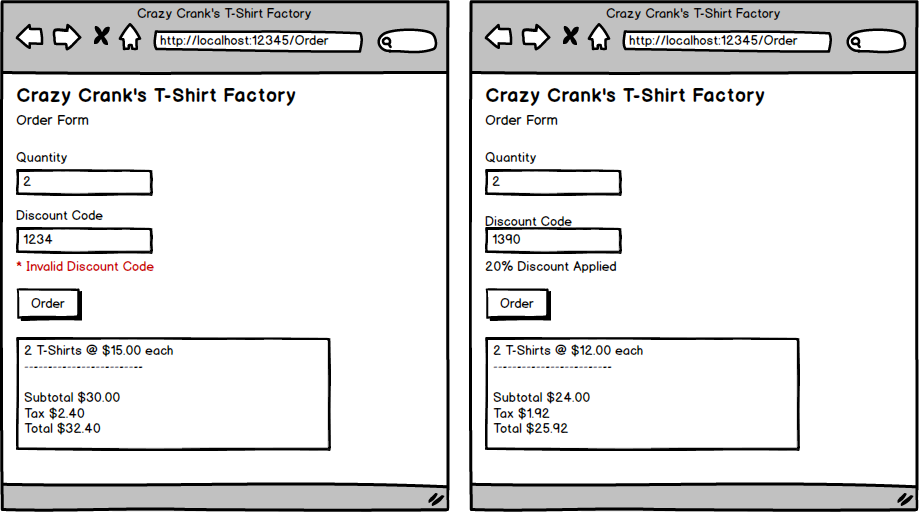
* Model class created and used \_\_\_\_\_ / 12pts
  + Model class named **DistanceConverterModel**
  + Model class has properties **DistanceInInches** and **DistanceInCentimeters**
  + Add the Distance as a required field, with a custom validation message, that must be between 1 and 500.
* Controller class created and implemented \_\_\_\_\_ / 18pts -3pts
  + Controller class named **DistanceConverterController**
  + Action method(s) named **Index**
  + **[HttpGet]** and **[HttpPost]** attributes applied to action methods
  + POST action method checks if the model is valid before performing conversions
  + POST action method performs the conversion correctly
  + Constant named **CM\_PER\_IN** defined and used
* View laid out as shown in the mockup \_\_\_\_\_ / 8pts
  + Title
  + Labels, Inputs, and Error Messages
  + Button
  + Output
* Output formatted as described \_\_\_\_\_ / 2pts
  + Inches and Centimeters are formatted with **two decimal places** in the View
* **Latest** version of Bootstrap used to style the page \_\_\_\_\_ / 10pts -1pt
  + Bootstrap stylesheet included via **CDN**
  + **<h1>** tag used for page header
  + **form-group** used to group labels, input boxes, and error messages
  + **form-control** class applied to input boxes
  + **text-danger** class applied to error messages
  + **btn** and **btn-primary** classes applied to button
  + **margins** applied to make the form look professional

**OrderForm [50 pts]**

Create a program for ordering T-Shirts.

* Users must enter a quantity.
* Users can enter an **optional** discount code.
* Shirts are normally $15 each. Sales tax is 8%.
* If the discount code is valid, then apply the relevant discount.
* If the discount code is not valid, then display an error message and calculate the receipt without a discount
* **See rubric on next page for more details.**

|  |  |
| --- | --- |
| **Discount Code** | **Discount Percentage** |
| 6175 | 30% off |
| 1390 | 20% off |
| BB88 | 10% off |



**OrderForm - Rubric**

* Model class created and used \_\_\_\_\_ / 12pts
  + Model class named **OrderForm**
  + Model class has properties **Quantity,** **DiscountCode, DiscountMessage, PricePerShirt, Subtotal, Tax, and Total**
  + **Quantity** has the **[Required]** attribute and **ErrorMessage** is customized
* Controller class created and implemented \_\_\_\_\_ / 18pts
  + Controller class named **OrderFormController**
  + Action method(s) named **Index**
  + **[HttpGet]** and **[HttpPost]** attributes applied to action methods
  + POST action method calculates the **subtotal, tax, and total** correctly
  + POST action method applies a valid discount code correctly to the **shirt cost**
  + POST action method calculates the subtotal, tax, and total correctly **even if a valid discount code is not provided**
* View laid out as shown in the mockup \_\_\_\_\_ / 8pts
  + Title and subtitle
  + Labels, Inputs, and Error Messages
  + Button
  + Output
* Output formatted as described \_\_\_\_\_ / 2pts -2pts
  + Monetary amounts are formatted as **currency**
  + Output looks like a receipt
  + Output has a thin black border
  + Output has a white background with black text
  + Output has a monospaced font
* Latest version of Bootstrap used to style the page \_\_\_\_\_ / 10pts
  + Bootstrap stylesheet included via **CDN**
  + **<h1>** tag used for page header
  + **form-group** used to group labels, input boxes, and error messages
  + **form-control** class applied to input boxes
  + **text-danger** class applied to error messages
  + **btn** and **btn-primary** classes applied to button
  + **margins** applied to make the form look professional