COMP123 – The Hero Class

Assignment 1
Implement the Hero Class

Due Week #4 (Friday January 30, 2015) @ midnight.

Value 5%

The Hero Class **Maximum Mark: 35**

Overview: Using MS Visual Studio, Build and Implement a **Hero** Class. The Hero class must include the following properties and methods:

Class
Hero
Properties (instance variables)
- strength : int - speed : int - health : int - name : string
Methods
+ generateAbilities(): void + fight(): void + hitAttempt(): bool + hitDamage(): int + show(): void

Instructions :

(25 Marks: Functionality, 6 Marks: Internal Documentation, 4 Marks: Version Control)

1. The Hero Class must include the following **Properties** and **Methods**: **(21 Marks: Functionality):**
    a. *Private* Properties: **strength** (int), **speed** (int), **health** (int) (3 Marks: Functionality).
    b. *Public* Properties: **name** (string) (1 Marks: Functionality).
    c. The **constructor Method** should take one parameter, **name** (string) and pass its value to the **name** property. It will also call the **generateAbilities** method (2 Marks: Functionality)
    d. A *private* **generateAbilities** Method randomly generates the ability numbers for the **strength**, **speed** and **health** properties. Each ability will be an integer between 1 and 100 (4 Marks: Functionality).
    e. A *public* fight Method calls the **hitAttempt** method. If **hitAttempt** returns true, then it will call the **hitDamage** method. The damage will then be **displayed** in a message on the Console (3 Marks: Functionality).

f. A *private* **hitAttempt** Method that will randomly determine if the Hero hits (this should be 20% of the time) otherwise it will return false. (3 Marks: Functionality).
g. A *private* **hitDamage** Method that calculates the damage the Hero causes to the target on a hit. The damage will be calculated by multiplying the Hero's strength property by a number between 1 and 6. The method will return this value. (3 Marks: Functionality).
h. A *public* **show** Method that will display the Hero's ability scores to the console (2 Marks: Functionality).

2. In your main method, **Implement** the Hero class by creating a new **hero** object. Have the hero call the **show** method to display his abilities. Then have the hero call the **fight** method. Please ensure to test your output thoroughly. (4 Marks: Functionality).
3. Include Internal Documentation for your program **(6 Marks: Internal Documentation):**
    a. Ensure you include a program header that indicates: the Author's name, Date last Modified, Program description, Revision History (4 Marks: Documentation).
    b. Ensure your program uses contextual variable names that help make the program human-readable (2 Marks: Documentation).
4. Share your files on **GitHub** to demonstrate Version Control Best Practices **(4 Marks: Version Control).**
    a. Your repository must include **your code** and be well structured (2 Marks: Version Control).
    b. Your repository must include **commits** that demonstrate the project being updated at different stages of development – each time a major change is implemented (2 Marks: Version Control).

SUBMITTING YOUR WORK

Your submission should include:

1. An external document (MS Word or PDF).
2. A link to your project files on GitHub.

Please zip all files in to a single project archive.

Program Code & Functionality

Technical Evaluation

Functionality	The program's deliverables are all met and the program functions as it should. No errors appear as a result of execution. User Input does not crash the program.	25
Internal Documentation & Readability	A program header is present and includes the name of the program, the name of the student, a short revision history and a short description of the program. All methods and classes include headers that describe their functionality and scope. Inline comments are used to indicate their function when code is new or unclear. Variable names are contextual wherever possible.	6
Version Control	**GitHub** is used to track App development. A **Commit history** will demonstrate the App being updated at regular points in time that correspond with the milestones of the project.	4

Creative Evaluation

		Mark
Creativity	The program's GUI / UI is attractive. The programmer has added additional elements outside of the scope of the program that enhance functionality, usability and fun.	0
	Total (/35)	35
	%	100.0%

This assignment is weighted **10%** of your total mark for this course.

Late submissions:

- 10% deducted for each additional day.

External code (e.g. from the internet or other sources) can be used for student submissions within the following parameters:

1. The code source (i.e. where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 10% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email.