Unit 6 Chapter 15 Assignment

Grading Information: This Program is **due** on **Date Specified**.

Comments are **REQUIRED**; flow charts and pseudocode are **NOT REQUIRED**.

Directions	**Points**
The files must be called <**LiFiUnit6Ch15.java**> **LiFiTime.java** (Time class file) The files must be called as specified above, (LiFi = Your Last Initial Your First Initial) *Proper coding conventions required the first letter of the class start with a capital letter and the first letter of each additional word start with a capital letter*. Only submit the **.java** files needed to make the program run. Do not submit the .**class** files or any other files.	5%
Style Components Include properly formatted prologue, comments, indenting, and other style elements as shown in Chapter 2 starting page 64 and Appendix 5 page 881-892.	5%
Topics covered in chapter Topics with * are covered in this assignment. *Exceptions and Exception Messages *try / catch *checked and unchecked exceptions *generic catch block *throws	
Basic Requirements Write a program that validates time when entered. It should output the time as either AM or PM or the error if entered incorrectly. See sample output below.	20%
LiFiUnit6Ch15.java • Driver class should loop until “q” is entered to quit • If enter is not “q”, then create an instance of the LiFiTime object passing the entry as an argument • If no error	10%

o Print time by calling print method in LiFiTime class • If error o Print error message (see sample)	
LiFiTime.java class Time objects should store the time in 2 integer instance variables, hour and minute, and include a string variable to hold the error. This should be initialized with null. LiFiTime Constructor: • Receive time as a string • Perform error checking to ensure time was entered in proper format to include a color (:) between the digits • Use indexOf and substring to separate the time string into the appropriate instance variables • Check hour to ensure is between 1 and 23 • Check minute to ensure is between 0 and 59 • If an error occurs, change the error instance variable to reflect the error (see sample) print method: • Include a separate method to print the date using printf to output time in digits for the hour and 2 digits for the minutes • Print AM if before noon and PM if after noon. Do NOT print in military time getError method: • Include a getError method that returns the error instance variable to the print method	30%
NOTE: Complete your activity and submit it by clicking “Submit Assignment”	
Total Percentage	100%

Sample
`Your output will vary based on the random numbers generated.`
Sample session (requires no user input):
`Enter time in the form mm:dd ("q" to quit): `*`5:20`*
`05:20 AM`
`Enter time in the form mm:dd ("q" to quit): `*`05:20`*
`05:20 AM`
`Enter time in the form mm:dd ("q" to quit): `*`05:80`*
`Invalid minute entered: 80`	

```
Enter time in the form mm:dd ("q" to quit): 15:20
03:20 PM
Enter time in the form mm:dd ("q" to quit): 25:20
Invalid hour entered: 25
Enter time in the form mm:dd ("q" to quit): 0520
Invalid separator entered: :
Enter time in the form mm:dd ("q" to quit): q
```