**44-542 Object Oriented Programming**

**Lab03: Classes Lab Activity**

**Objective:** Covers the creation and usage of **Concrete class** and **Scanner** class.

**NOTE:**

* Do not hard code any values.
* Do not use any looping statements for this lab.
* Check the sample output to know how the results need to be printed.
* Read every instruction carefully and follow them strictly.
* Do not change the name of the attributes and methods given below.
* This lab contains two parts (Part 1 & Part 2). Make sure you complete both the parts.
* @author notation should contain your full name.
* Generate the java docs for your project.

**Part 1**

1. Create a New Project and name it as **Lastname\_Lab03Classes** where **Lastname** is your last name.
2. Create a new package in the project created and name it as **stores**.
3. Create a new Java class in **stores** package and name it as **Store**.
4. Write statements to declare the following attributes. Do not add any instance variables beyond those shown here. Access specifiers must be private for all the given instance variables.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Type** | **Attribute Description** |
| **storeName** | **String** | Name of the store (Note: For this project, we always consider the name of the store has exactly three words.  Examples: Reliance Fresh Walmart (or)  Macys Private Ltd.) |
| **storeID** | **int** | Unique identifier of the store |
| **phoneNumber** | **String** | This is a ten-digit phone number with only numerical values. |
| **street** | **String** | Name of the street where the store is located (This input for this variable should also contain the house number along with the street name  Ex: 654 West Addison Street) |
| **city** | **String** | Name of the city where the store is located |
| **state** | **String** | Name of the state where the store is located (Note: The input for this variable should contain the full state name not the state code) |
| **zipCode** | **int** | This is a five-digit zip code of the state where the store is located |

1. Constructors:
   1. Create a constructor with parameters. The parameters are used to set the values of the instance variables.

**public Store(String storeName, int storeID, String phoneNumber, String street, String city, String state, int zipCode)**

* 1. Create a no-argument constructor with no body.

1. Methods:
   1. Write the getter and setter methods for each of the instance variable declared.
   2. Write a method **getFormattedAddress()** to return the address of the store. Look, the example given below to know the format how it should return the address of the store. For example:

108 N State Street

Chicago, Illinois-60018

* 1. Write a method **getFormattedPhoneNumber()** to return the phone number in this format (ddd)ddd-dddd where d is the numerical digit.
  2. Write a method **replaceStoreName(String oldName, String newName).** This method should replace the old store name with the new store name. Return type of this method should be void.
  3. Write a method **getFirstWordOfStoreName()**to return the first word in the store name.
  4. Write a method **getMiddleWordOfStoreName()**to return the middle word in the store name.
  5. Write a method **getLastWordOfStoreName()**to return the last word in the store name.
  6. Override the **toString()** method which should be used to display the object. This method should return a String. Please see the sample output to know the pattern.

(Note: This method should use **getFormattedAddress(), getFormattedPhoneNumber()**for printing the address and phone number of store)

1. Create a new Java Main class in **stores** package and name it as **StoreDriver**.

|  |
| --- |
| **StoreDriver Class Specifications**  **NOTE: Make sure your output matches with the sample output given.**  1. Print the message “Testing the Store Class”.  2. Declare and create a **Store** object as **store1** with parameterized constructor with  values "Timmy Tommy Enterprises", 1000, "3127869900", "311 Jarvis Square",  "Chicago", "Illinois", 60018 respectively.  3. Print the message “Testing the Getter methods or Accessors of Store Class”.  4. Invoke all the getter methods on **store1** and print the result.  5. Print the message “Testing the toString method”.  6. Invoke the **toString()** method on **store1** and print the result.  7. Print the message “Testing the user defined methods”.  8. Invoke the **getFirstWordOfStoreName()** method on **store1** and print the result.  9. Invoke the **getMidddleWordOfStoreName()** method on **store1** and print the  result.  10. Invoke the **getLastWordOfStoreName()** method on **store1** and print the result.  11. Invoke the **replaceStoreName()** method on **store1** to replace the existing store  name with the “JC Penny Store”.  12. Print the message "Printing the store1 object after invoking the replace method”.  13. Invoke the **toString()** method on **store1** and print the result.  14. Declare and create a new **Store** object as **store2** using no-arg constructor.  15. Print the message “Testing the Getter methods or Accessors of Store Class with no-arg  constructor”.  16. Invoke all the getter methods and print the result.  17. Invoke the **setStoreName()** method on **store2** with the value “KC India Mart”.  18. Invoke the **setStoreID()** method on **store2** with the value 1001.  19. Invoke the **setPhoneNumber()** method on **store2** with the value “9136818080”.  20. Invoke the **setStreet()** method on **store2** with the value “8542 w 133rd Street”.  21. Invoke the **setCity()** method on **store2** with the value “Overland Park”.  22. Invoke the **setState()** method on **store2** with the value “Kansas”.  23. Invoke the **setZipcode()** method on **store2** with the value “66213”.  24. Print the message “Testing the store class using toString after invoking the Setter  methods or Mutators”.  25. Invoke the **toString()** method on **store2** and print the result.  26. Print the message “Testing the Scanner class to take input from the console”.  27. Declare and create a **Scanner** object as **scannerObject** to take input from the  console.  28. Print the messages given in the sample run accordingly.  Here is the sample run of the program how the input should be given from the console  Enter the store details  Enter the store name:  **Dunkin Donuts Limited**  Enter the store ID:  **1003**  Enter the store phone number:  **6034153654**  Enter the street name of the store:  **656 Suncook Valley Hwy S**  Enter the city name, state name, zip code of the store in one  line without any commas:  **Pittsfield Massachusetts 01201**  29. Declare and create **Store** object **store3** with parameterized constructor  with the values you have taken from the console. .  30. Print the message “Testing the toString method without invoking the method”.  30. Print the **store3** object and print the result. Do not invoke the **toString()** method |

|  |
| --- |
| **Sample Output Window**  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Testing the Store Class\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Testing the Getter methods or Accessors of Store Class  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Store Name: Timmy Tommy Enterprises  Store ID: 1000  Street of the Store: 311 Jarvis Square  City of the Store : Chicago  State of the Store: Illinois  Zip code of the Store: 60018  Phone number of the Store: 3127869900  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Testing the toString method  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Timmy Tommy Enterprises (1000)  311 Jarvis Square  Chicago, Illinois-60018  (312)786-9900  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Testing the user defined methods  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  First word of store name: Timmy  Middle word of the store name: Tommy  Last word of the store name: Enterprises  Printing the store1 object after invoking the replace method  JC Penny Store (1000)  311 Jarvis Square  Chicago, Illinois-60018  (312)786-9900  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Testing the Getter methods or Accessors of Store Class with no-arg constructor  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Store Name: null  Store ID: 0  Street of the Store: null  City of the Store: null  State of the Store: null  Zip code of the Store: 0  Phone number of the Store: null  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Testing the store class using toString after invoking the Setter methods or Mutators  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  KC India Mart (1001)  8542 w 133rd Street  Overland Park, Kansas-66213  (913)681-8080  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Testing the scanner class to take input from the console  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Enter the store details  Enter the store name:  Dunkin Donuts Limited  Enter the store ID:  1003  Enter the store phone number:  6034153654  Enter the street name of the store:  656 Suncook Valley Hwy S  Enter the city name, state name, zip code of the store in one line without any commas:  Pittsfield Massachusetts 01201  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Testing the toString method without invoking the method  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Dunkin Donuts Limited (1003)  656 Suncook Valley Hwy S  Pittsfield, Massachusetts-1201  (603)415-3654 |

**Part 2**

**Scenario:** Assume that you are a software developer in an IT company. Your task is to implement the GPA calculator for a higher education institution for the following requirements.

* + The system shall be able to calculate the GPA of a student by taking the following details only.
    - Student first name
    - Student last name
    - Student identification number
    - Letter grades for only three courses (Assume that every course is of 3 credit hours. Letter grades include A or B or C or D or F )
  + The system shall be able to prompt the user to enter the first name, last name, student identification number, and the letter grades for the three courses
  + The system shall be able to print the calculated GPA with appropriate labels. You have the freedom to choose your format to print the labels and result.
  + The system shall test the logic of GPA calculation on some dummy data given the user from the keyboard.

As a developer, you should include your formula for calculating GPA in the comments of the source code. If you have any questions in implementing this project, please ask your team leads (teaching assistants) for aspects that are more technical.

**Note: You should create a new package in the same project you created for this lab.**

**Submit you solution by following the steps below:**

* Save your files in **NetBeans**.
* Zip your entire Project. (It should be called ***Lastname*\_Lab03Classes.zip** where **Last name** is your last name.)
* Submit the Zip file to the **Lab03Classes** drop box.
* Download the Zip file you have submitted.
* Look in the Zip file and verify the class files in the Zip folder are correct. If not resave your project in **NetBeans** and resubmit.