



Object Orientated Programming

Lab Manual 3



Instructor:

Mr. Samyan Qayyum Wahla

Learning Objectives:

- Student should be able to create class using java
- Learn to write well documented and formatted code.

CLO:

- CLO1

Registration Number:

Name:

Guidelines/Instructions:

- Use of Notepad is must in this lab.
- Write well commented code.
- Name of variables should be meaningful.
- Use camel case for the naming convention
- Code should be well formatted
- Create meaningful variable names. Add comments for readability. Indent each line of your code.
- Plagiarism/Cheating is highly discouraged by penalizing to both who tried and one who shared his/her code.

Reading Content:

- Lecture content of the week 3.

Task

Write a Java program in which you are required to define a class named **Student**. The class must include the following data members.

- **studentName** //data type should be String (name should be alphabetic, special characters and numbers are not allowed)
- **registrationNumber** //data type should be String (Format should be like this: 2020-CS-888, any other format should be handled in setter function)
- **dateOfBirth** //data type should be Date (Date of birth should be less than 1st January 2005 and greater than 31st December 1990, add conditions for invalid data)
- **cgpa** // data type should be double (should be between 0 and 4 inclusive)
- **cnic** //data type should be String (only numbers are allowed with valid CNIC)
- **numberOfInstances** //This attribute should track the number of objects of the class Student. (Make it static. Don't create setter for this attribute)

1. Your Program should define two constructors for the class **Student**
 - a constructor with **no parameter**
 - a constructor with **two parameters** (Name, Registration Number)
 - All of these constructors are meant to initialize their respective objects.
2. Define getter setter for each data member
3. Apart from getter, setter and constructor, define the following function

- i. **String getAge()** – it should calculate age of student upto current date format given in output section.
 - ii. **String getStatus()** – function should tell the status of student in the following cases
 - a. IF CGPA less than 2.0 - Status is Suspended
 - b. IF CGPA between 2.0 and 2.5 - Status is Below Average
 - c. IF CGPA between 2.5 and 3.3 - Status is Average
 - d. IF CGPA between 3.3 and 3.5 - Status is Below Good
 - e. IF CGPA greater than 3.5 – Status is Excellent
 - iii. **int numberOfWordsInName()** – it should return number of words in students name
 - iv. **String getGender()** – if last digit of CNIC is odd then it should return MALE otherwise FEMALE
 - v. **String toString()** – Display the output in specific format given in output
 - vi. **void input()** – Take the input from the user in this method. You can read input from user using InputDialog or Scanner.
4. Declare two objects (1 for each type of constructor) in **Driver class**.
 5. Use the javadoc comments to document your both classes.

Output

Your output should be like the following with same formatting

Name: Sarfraz Ahmed (Contain 2 words)
Registration Number: 2018-CS-999
CGPA: 2.9 Average
Date of Birth: December 27, 1999 (Age is 21 years 8 months and 12 days)
CNIC: 3567912356781
Gender: Male

Name: Hafiz Muhammad Aslam (Contain 3 words)
Registration Number: 2017-CS-999
CGPA: 3.4 Good
Date of Birth: November 27, 1999 (Age is 21 years 9 months and 12 days)
CNIC: 3567912356785
Gender: Male

Student class has the 2 objects

What to submit

You are simply required to submit a source file (**Student.java** and **Driver.java**(in which main function is defined)) that includes the implementation of the above mentioned program. No extra file should be submitted.