

# OBJECT ORIENTED PROGRAMMING (JAVA) (CST8284)

#### LAB 3

# Constructors Overloading and Chaining



#### Introduction

- The essence of this lab is to work with constructors and to show overloaded constructor calls.
- It demonstrates chaining overloaded constructors using "this" keyword.
- It helps to consolidate understanding of the chaining rules learned in the theory class
- This program also displays a good example of writing comments in your code using the Javadoc style.



# Preparing for your Tasks...

- Take some time to review the concept of constructors and constructor chaining studied at the theory class especially the rules.
- Review all resources provided to you for the lab.
- For this learning exercise, the order of implementation for both 'upwards' and 'downwards' chaining will be applied and the sequence will be described for you.



#### What You Would Demonstrate

- Show your Professor the updated version of the code provided to you.
- Execute your code to show that it has no errors and that the output is correct (review the sample output provided to you).
- Answer your professor's questions.
- Show Javadoc output file capturing the comments in your code





## **Description**

- An extensive description has been provided for you in each class.
- There are three Java code files (classes) and one sample output file provided namely:
  - Car.java
  - Bicycle.java
  - TestDemo.java
  - SampleOutput



#### TASKS: You are required to...

- Create a new project in Eclipse Lab 3
- Follow the instructions on the code files given:
  - Car.java
  - Bicycle.java
  - > TestDemo.java
  - Sample output the output example
- Create a package and load the code files. Read to clearly understand the sequence.
- Do NOT load the sample output file.



# You are required to...(2)

Check the code files to identify the TO DO portions in the code.

Update the commented sections in each code file as required.

The chaining of constructors in the Car class and Bicycle class should be implemented in a specific order (upwards / downwards) for your learning purposes.



#### **Updating your code...**

- Ensure that you understand the program and what you have been asked to do.
- Update only sections specified (check comments in the code)
- Review the resources posted for help
- Run your code and make sure it executes without errors.
- Ensure that your output corresponds to the pattern of the SampleOutput file provided.



### Ready to demo your work...

- Point out to your professor what you did.
- Answer questions based on your work.
- Generate and show the comments in Javadoc (including the ones you added).



#### **Rubrics**

- Correct, complete implementation and execution of the TO DO Tasks 1.5%
- Javadoc comments <u>added</u>, generated and displayed – 0.5%
- Correct answering of questions regarding the code – 1%



#### References

❖ Big Java Early Objects, 7/E. Author: Horstmann, C. Wiley. ISBN: eText: 978-1-119-49909-1 or loose-leaf paper: 978-1-119-74020-9.

