Department of Computer and Software Engineering – ITU

CE101T: Object Oriented Programming

Course Instructor: Nadir Abbas	**Dated: 16/02/2025**
Teaching Assistant: Zainab, Nahal, Bilal	**Semester: Spring 2025**
Session: 2024-2028	**Batch: BSCE2024**

Assignment 4. Introduction To Classes, Constructors, Destructors, Inline Functions, and Static Variables

Name	Roll number	Obtained Marks/35

Checked on: _____

Signature: _____

Submission:

- Email instructor or TA if there are any questions. You cannot look at others' solutions or use others' solutions, however, you can discuss it with each other. Plagiarism will be dealt with according to the course policy.
- Submission after due time will not be accepted.

In this assignment you have to do following tasks:

Task 1: Ensure that you have installed all three softwares in your personal computer (Github, Cygwin & CLion). Now, accept the assignment posted in classroom (e.g Google, LMS etc) and after accepting, clone the repository to your computer. Make sure you have logged into the github app with your account.

Task 2: Open Cygwin app, Move to your code directory with following command "cd <path_of_folder>"
<path_of_folder> can be automatically populated by dragging the folder and dropping it to the cygwin window.
Run the code through Cygwin, use command "make run", to get the output of the code

Task 3: Solve the given problems, write code using **CLion** or any other IDE.
Task 4: Keep your code in the respective git cloned folder.
Task 5: Commit and Push the changes through the Github App
Task 5: Write the code in separate files **(as instructed)**. Ensure that file names are in lowercase (e,g **main.cpp**).

Task 6: Run '**make run**' to run C++ code
Task 7: Run '**make test**' to test the C++ code

Question 1: Class Definitions

Define the following two classes separately.

1. **Class: Car**
   Members:

   - string **brand** (Car brand)
   - string **model** (Car model)
   - int **year** (Manufacturing year)

   Constructors:
   - Default Constructor: Initializes brand as "", model as "", and year as 0.
   - Parameterized Constructor: Takes values for brand, model, and year.
   - Copy Constructor: Initializes a new Car object using an existing Car object.
   - Destructor: Prints a message when a Car object is destroyed.

2. **Class: Laptop**
   Members:

   - string **brand** (Laptop brand)
   - string **processor** (Processor type)
   - int **ram** (RAM size in GB)

   Constructors:
   - Default Constructor: Initializes brand as "", processor as "", and ram as 0.
   - Parameterized Constructor: Takes values for brand, processor, and ram.
   - Copy Constructor: Initializes a new Laptop object using an existing Laptop object.
   - Destructor: Prints a message when a Laptop object is destroyed.

Question 2: Inline Member Function for Car

Implement the following **inline** member function in the Car class:

- **bool isClassicCar():** Returns true if the car is older than 25 years from the current year, otherwise false.

Question 3: Inline Member Function for Laptop

Implement the following **inline** member function in the Laptop class:

- **bool isHighPerformance():** Returns true if the laptop has ram greater than or equal to 16 GB, otherwise false.

Question 4: Getter and Setter Functions for Car

Implement **getter** and **setter** functions for each member of the Car class:

- **void setBrand(string b)**
- **string getBrand()**
- **void setModel(string m)**
- **string getModel()**
- **void setYear(int y)**
- **int getYear()**

Question 5: Getter and Setter Functions for Laptop

Implement **getter** and **setter** functions for each member of the Laptop class:

- **void setBrand(string b)**
- **string getBrand()**
- **void setProcessor(string p)**
- **string getProcessor()**
- **void setRam(int r)**
- **int getRam()**

Question 6: Static Variable for Car Count

- Declare a **static integer carCount** the class.
- Modify the constructors and destructor to update carCount whenever a Car object is created or destroyed.
- Implement the function **static int getCarCount()** that returns the current value of carCount .

Question 7: Static Variable for Laptop Count

- Declare a **static integer laptopCount** the class.
- Modify the constructors and destructor to update laptopCount whenever a Laptop object is created or destroyed.
- Implement the function **static int getLaptopCount()** that returns the current value of laptopCount.

Question 8: Copy Constructor in Action for Car

- Write a function **Car cloneCar(const Car& original)** that takes a Car object as a parameter and creates a copy using the copy constructor and returns it.

Question 9: Copy Constructor in Action for Laptop

- Write a function **Laptop cloneLaptop(const Laptop& original)** that takes a Laptop object as a parameter and creates a copy using the copy constructor and returns it.

Question 10: Combining Concepts

- Create a function **Car compareCars(Car& car)** that compares itself with the given car object and returns which car is older.
- Create a function **Laptop compareLaptops(Laptop& laptop)** that compares itself with the given Laptop object and returns which laptop has more RAM.

Assessment Rubric for Assignment

Performance metric	CLO	Able to complete the task over 80% (4-5)	Able to complete the task 50-80% (2-3)	Able to complete the task below 50% (0-1)	Marks
1. Realization of experiment	3	Executes without errors excellent user prompts, good use of symbols, spacing in output. Through testing has been completed.	Executes without errors, user prompts are understandable, minimum use of symbols or spacing in output. Some testing has been completed.	Does not execute due to syntax errors, runtime errors, user prompts are misleading or non- existent. No testing has been completed.	
2. Conducting experiment	2	Able to make changes and answered all questions.	Partially able to make changes and few incorrect answers.	Unable to make changes and answer all questions.	
3. Computer use	4	Document submission timely.	Document submission late.	Document submission not done.	
4. Teamwork	4	Actively engages and cooperates with other group member(s) in effective manner.	Cooperates with other group member(s) in a reasonable manner but conduct can be improved.	Distracts or discourages other group members from conducting the experiment	
5. Laboratory safety and disciplinary rules	2	Code comments are added and does help the reader to understand the code.	Code comments are added and does not help the reader to understand the code.	Code comments are not added.	
6. Data collection	2	Excellent use of white space, creatively organized work, excellent use of variables and constants, correct identifiers for constants, No line-wrap.	Includes name, and assignment, white space makes the program fairly easy to read. Title, organized work, good use of variables.	Poor use of white space (indentation, blank lines) making code hard to read, disorganized and messy.	
7. Data analysis	3	Solution is efficient, easy to understand, and maintain.	A logical solution that is easy to follow but it is not the most efficient.	A difficult and inefficient solution.	
Total (out of 35):					