CS3201: OBJECT ORIENTED PROGRAMMING LABORATORY

Topic: Pointers, Constructors, Destructors, Classes and Objects

Date: 12/02/2024

OBSERVATION

1. What happens when one does not define a destructor within a class?
2. Predict the output:

    ```
    #include <iostream>

    using namespace std;

    int main() {

            int a = 32, *ptr = &a;

            char ch = 'A', &cho = ch;

            cho += a;

            *ptr += ch;

            cout << a << ", " << ch << endl;

            return 0;

    }
    ```

    (a) 32,A   (b) 32,a   (c) 129,A   (d) 129, a

3. Explain what the & operator does in the two cases described below-
    e.g. int *ptr = &x;
    e.g. int &x = y;

4. Does the following code have errors? If so, fix them. What would be the output?

    ```
    void fun(int *p)
    {
      int q = 10;
      p = &q;
    }

    int main()
    {
      int r = 20;
      int *p = &r;
      fun(p);
      printf("%d", *p);
      return 0;
    }
    ```

EXECUTION QUESTIONS

1. Consider a class `Car` representing various attributes and behaviors of cars. Implement the class with the following specifications:
   - It should have private member variables `brand` and `mileage`.
   - Implement a static member variable `totalCars` to keep track of the total number of cars created.
   - Implement a constructor that initializes `brand` and `mileage` and increments `totalCars`.
   - Implement a destructor that decrements `totalCars` when an object is destroyed.
   - Implement a member function `displayTotalCars()` that displays the total number of cars created.

   Now, create instances of the `Car` class and perform the following operations:

   - Create three car objects with different brands and mileage.
   - Display the total number of cars created using the `displayTotalCars()` member function.

2. Implement a C++ program to create a game. In this game, the player controls a pointer inside an integer array. The player's objective is to move the pointer left or right within the array to reach the target value.

   In this game:

   - The player controls a pointer within an array of integers.

   - The player's goal is to move the pointer left or right to reach a target value within the array.

   - The player enters 'L' to move the pointer left and 'R' to move it right.

   - The game continues until the pointer reaches the target value.

   - After each move, the array and the current pointer position are displayed.

   >>If array inputted by the user during program run is : 5 6 7 9 4 0

   Sample O/p:
   Enter the target value: 7
   [5] 6 7 9 4 0
   Move left (L) or right (R): R
   5 [6] 7 9 4 0
   Move left (L) or right (R): R

   Congratulations! You have reached the target value.