Object Oriented Programming LAB – BSDSF24

(Both Morning and Afternoon)

Lab 04 –25-02-2025

use notepad++ and developer command prompt for the following tasks

1. Correct and Complete the following class **robot. [50 marks]**

```
class robot
{
public:
        string getRobotName() const {}
        void setRobotName(string n) {}
        int getLocationX() const {}
        void setLocationX(int v) {}
        int getLocationY() const {}
        void setLocationY(int v) {}
        char getDirection() const {}
        void setDirection (char d) {}
        void setRobot(string n, int cx, int cy, char d) {}
        bool canStep(){}
        void takeStep(){}
        void turnLeft (){}
        void turnRight (){}
        void turnBackword (){}
        void display() const {}
private:
        string name;
        int row, col;
        char direction; // u(up), d(down), l(left), r(right) on Cartesian plane
};
```

canStep function returns true only when front of ROBOT is clear, takeStep will change the current location of the robot to location ahead it, if possible). turn???? functions will change the direction of the robot w.r.t current direction, i.e. if a robot current direction is l (left), turnLeft() will change its direction to d (down). display() will output a robot state as

R1 is at (x,y) towards d

2. Testing the class **robot [15, 15, and 20 marks]**.

a. After creating the robot class, in **main** logic test the functions for yourself. Note that, *in main logic, each created **robot** must have unique name. Each **robot** can be placed and can move on a board on size 100 X 100 (means **robot** location can vary from 0,0 to 99,99).*

b. Create an array of **six** robots. Name them, R, S, T, Q, P and Z, and place them at different locations and in different directions of your own choice. Then display the location of each robot. Move each robot 4 times (using takeStep function) on board in their direction (if possible) and display their modified location.

c. Now, you have to move robots R, S, Q, P and Z towards robot T. Here you have to use turn???? and takeStep functions provided that at any instant of time (an iteration of the loop), each robot takes only one action. You must display a notification as soon as a robot approaches at ***adjacent*** location to T or to some **robot** arrived there earlier. Except, robot member functions call, everything must be handled in main logic.

Thank you for your **patience**