Names: **Aarushi Roychoudhury (ID: 30113987)**
And Farica Mago (ID: 30111924)
Course Name: **Principles of Software Design**
Lab Section: **B01**
Course Code: **ENSF 480**
Assignment Number: **Lab 4**
Submission Date and Time: **2022-10-21**

Exercise A

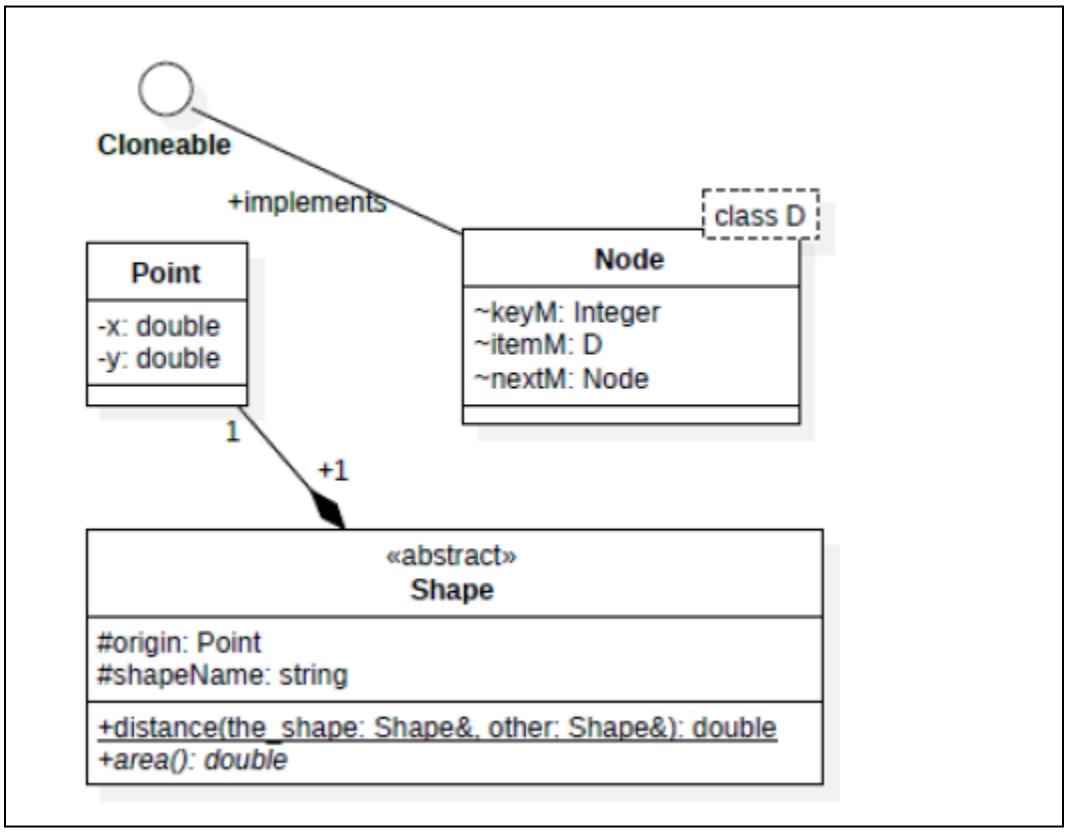
Exercise B

Exercise C

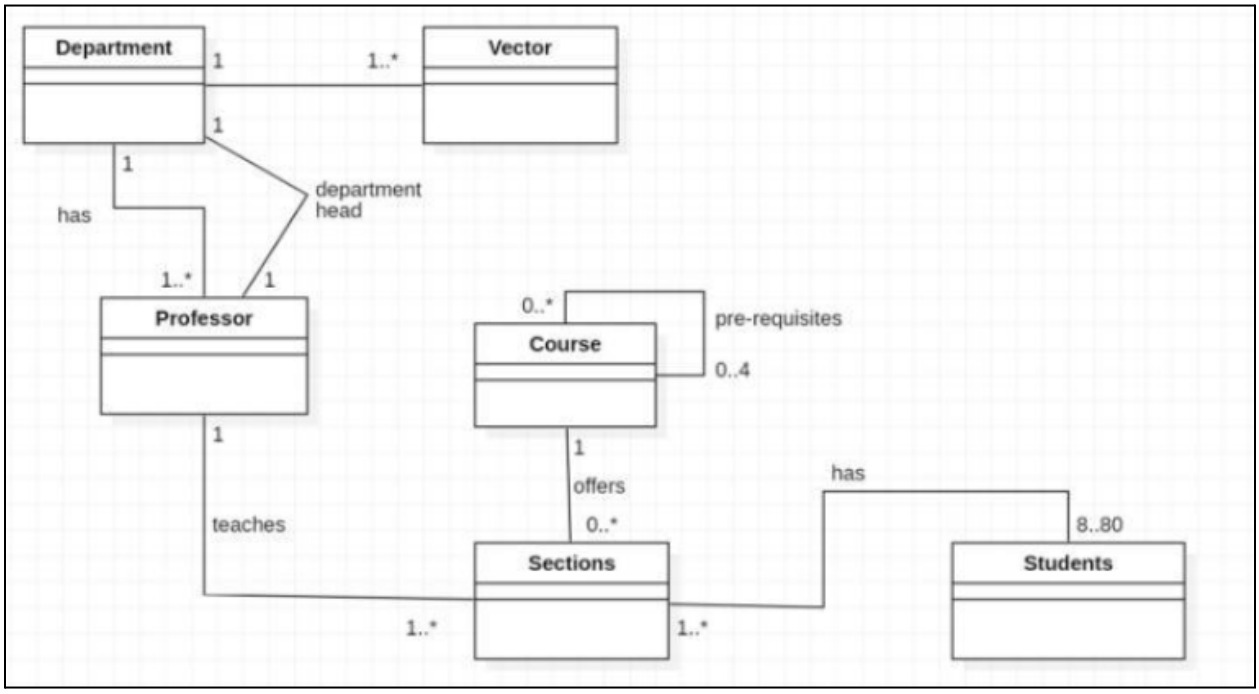
Exercise D

Exercise E

Exercise A


Cloneable
+implements
class D
Node
~keyM: Integer
~itemM: D
~nextM: Node
Point
-x: double
-y: double
1
+1
«abstract»
Shape
#origin: Point
#shapeName: string
+distance(the_shape: Shape&, other: Shape&): double
+area(): double

Exercise B



Exercise C

Car.h

```
#include "Vehicle.h"

class Car: public Vehicle{
    private:
        int seats;
    public:
        Car(int seats);
        void turn();
}
```

Moveable.h

```
class Moveable{
    public:
        virtual void forward() = 0;
        virtual void backward() = 0;
};
```

Resizable.h

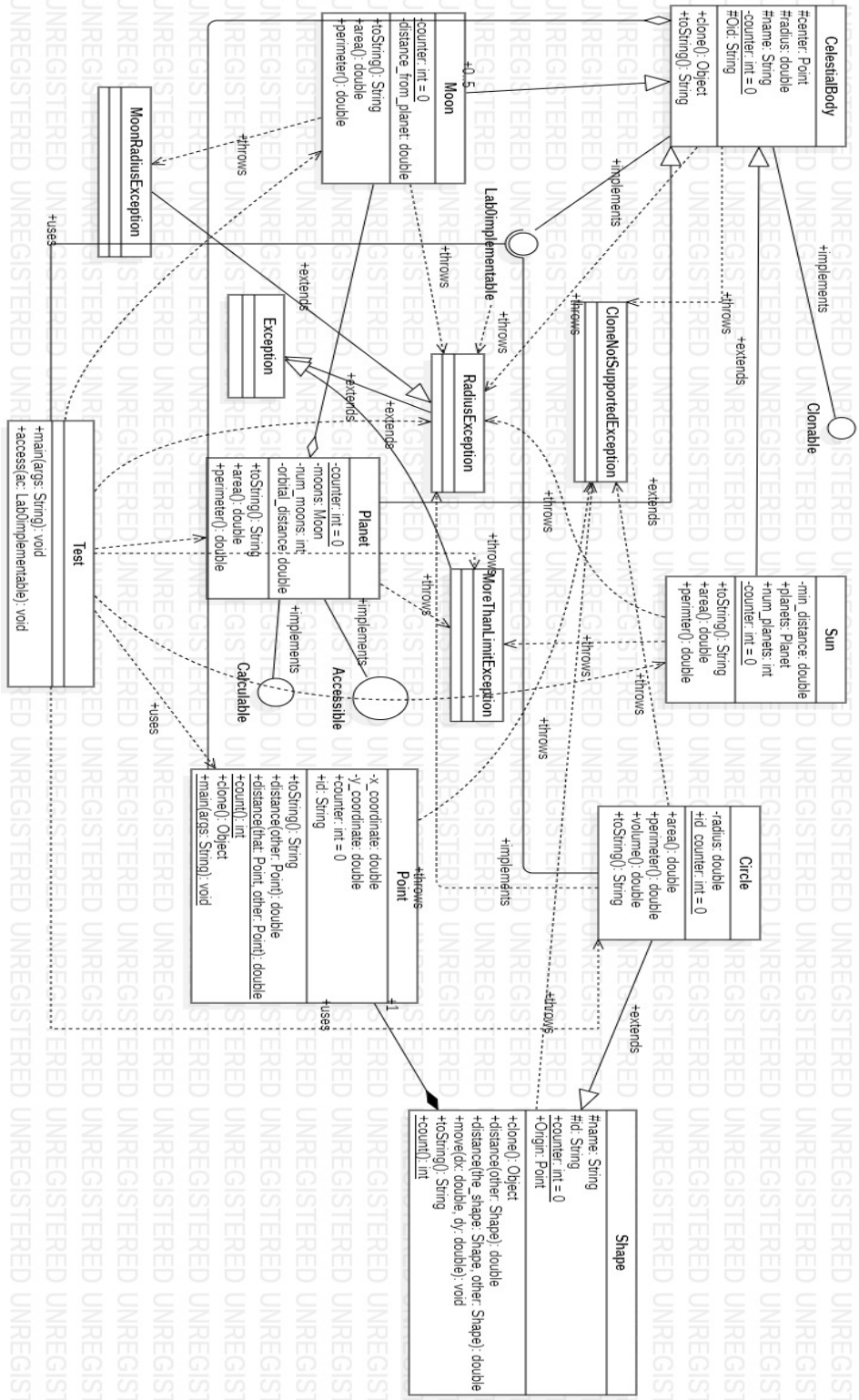
```
class Resizable{
    public:
        virtual void enlarge(int n) = 0;
        virtual void shrink (int n) = 0;
};
```

Vehicle.h

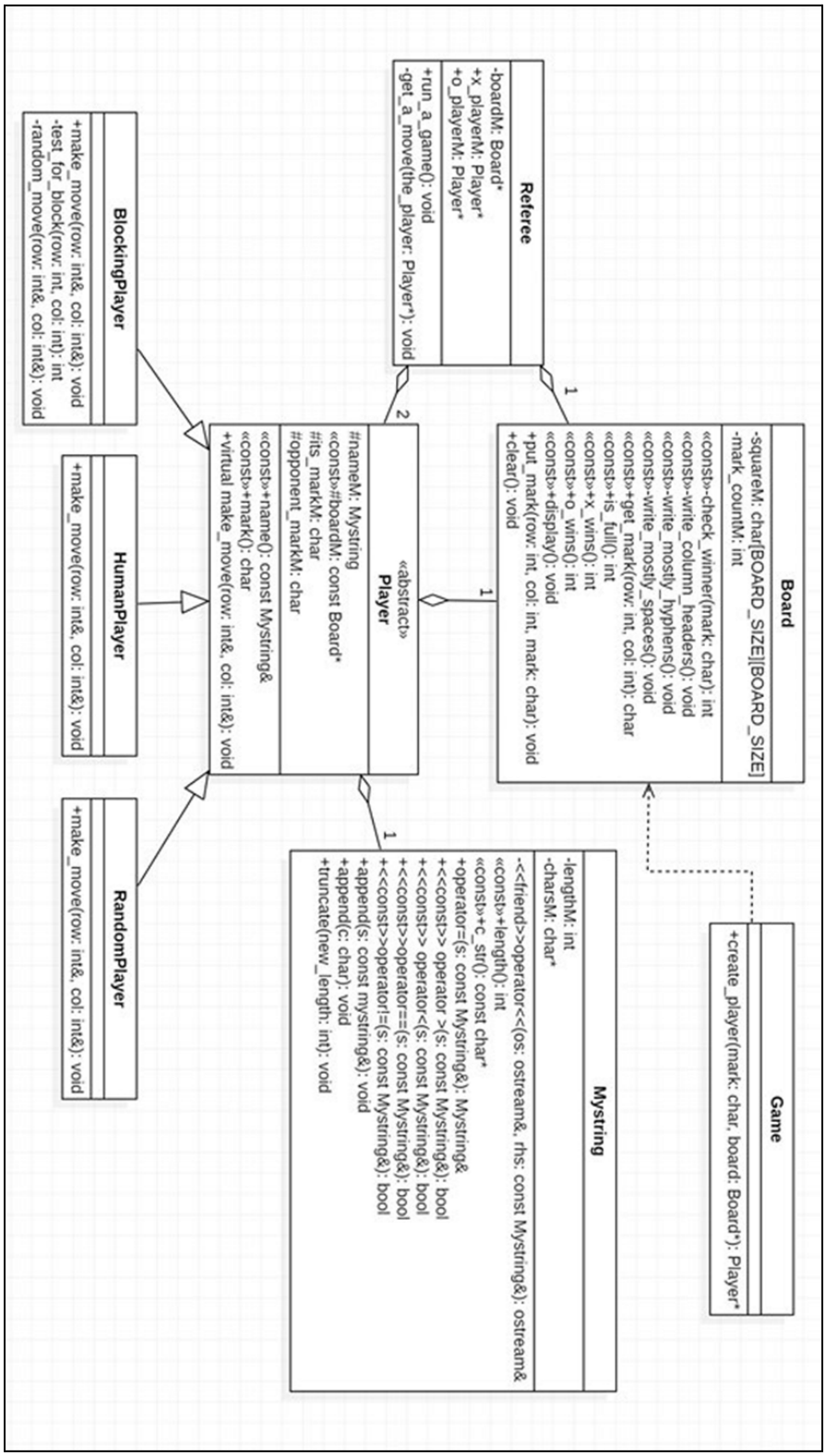
```
#include <string>
#include "Moveable.h"

class Vehicle : public Resizable, public Moveable{
    protected:
        std::string name;
    public:
        Vehicle(std::string name);
        void move();
};
```

Exercise D


CelestialBody
#center: Point
#radius: double
#name: String
-counter: int = 0
#Oid: String
+clone(): Object
+toString(): String
Clonable
Moon
-counter: int = 0
-distance_from_planet: double
+toString(): String
+area(): double
+perimeter(): double
0..5
Sun
-min_distance: double
+planets: Planet
+num_planets: int
-counter: int = 0
+toString(): String
+area(): double
+perimiter(): double
Circle
-radius: double
#id: counter: int = 0
+area(): double
+permmeter(): double
+volume(): double
+toString(): String
Shape
#name: String
#id: String
+counter: int = 0
+Origin: Point
+clone(): Object
+distance(other: Shape): double
+distance(the_shape: Shape, other: Shape): double
+move(dx: double, dy: double): void
+toString(): String
+count(): int
Point
-x_coordinate: double
-y_coordinate: double
+counter: int = 0
+id: String
+toString(): String
+distance(other: Point): double
+distance(that: Point, other: Point): double
+count(): int
+clone(): Object
+main(args: String): void
Planet
-counter: int = 0
-moons: Moon
-num_moons: int
-orbital_distance: double
+toString(): String
+area(): double
+perimeter(): double
Test
+main(args: String): void
+access(ac: LabOImplementable): void
LabOImplementable
Accessible
Calculable
Exception
RadiusException
MoreThanLimitException
CloneNotSupportedException
MoonRadiusException
+implements
+extends
+throws
+uses
+1
+0..5

Exercise E


Board
-squareM: char[BOARD_SIZE][BOARD_SIZE]
-mark_countM: int
«const»-check_winner(mark: char): int
«const»-write_column_headers(): void
«const»-write_mostly_hyphens(): void
«const»-write_mostly_spaces(): void
«const»+get_mark(row: int, col: int): char
«const»+is_full(): int
«const»+x_wins(): int
«const»+o_wins(): int
«const»+display(): void
+put_mark(row: int, col: int, mark: char): void
+clear(): void
Game
+create_player(mark: char, board: Board*): Player*
Referee
-boardM: Board*
+X_playerM: Player*
+O_playerM: Player*
+run_a_game(): void
-get_a_move(the_player: Player*): void
«abstract»
Player
#nameM: Mystring
«const»#boardM: const Board*
#its_markM: char
#opponent_markM: char
«const»+name(): const Mystring&
«const»+mark(): char
+virtual make_move(row: int&, col: int&): void
Mystring
-lengthM: int
-charsM: char*
-<<friend>>operator<<(os: ostream&, rhs: const Mystring&): ostream&
«const»+length(): int
«const»+c_str(): const char*
+operator=(s: const Mystring&): Mystring&
+<<const>> operator >(s: const Mystring&): bool
+<<const>> operator<(s: const Mystring&): bool
+<<const>>operator==(s: const Mystring&): bool
+<<const>>operator!=(s: const Mystring&): bool
+append(s: const mystring&): void
+append(c: char): void
+truncate(new_length: int): void
BlockingPlayer
+make_move(row: int&, col: int&): void
-test_for_block(row: int, col: int): int
-random_move(row: int&, col: int&): void
HumanPlayer
+make_move(row: int&, col: int&): void
RandomPlayer
+make_move(row: int&, col: int&): void
1
2
1
1
1