BITS PILANI, DUBAI CAMPUS

DUBAI INTERNATIONAL ACADEMIC CITY, DUBAI

**FIRST SEMESTER 2024 – 2025**

**COURSE:** F213 (Object Oriented Programming)

**COMPONENT:** Lab 12 **DATE:** 20th November 2024

**INTERFACES**

**Q.1** Develop a JAVA program to simulate the movement of chess pieces on a chess board, using interfaces. Define an interface movable with a method move(String newPos). Further define an abstract class Piece, and classes for each type of chess piece (eg King, Queen etc) and override the method move(String newPos). (Please refer to the sample implementation shown below)

interface Movable() { void move(String newPos); } abstract class ChessPiece implements movable {

String name; // eg “King”, “Queen” etc.

String color; // “White”, “Black”

String curPos; // “a1”,“c5” std chess board naming convention public ChessPiece (/\*appropriate parameters \*/ ) { }

}

class King extends ChessPiece {

/\* appropriate implementation details \*/

}

In the main program create an ArrayList of Pieces, (any 3 of your choice) and simulate the movement on these pieces on the chess board. You need not check if the move is a legal move, just print the message <Color> <PieceName> moving from <curPos> to <newPos> as shown below.

Eg: (Comp) : Created 3 chess Pieces (eg. King(White):1, Queen(Black):2 , Pawn(White):3). (Comp) : Enter which piece to be moved (0 to Quit) & new Position

(User) : 1 a6

(Comp) : White King moving from a7 to a6

(Comp) : Enter which piece to be moved (0 to Quit) & new Position (User): 2 c5

(Comp) : Black Queen moving from c2 to c5

(Comp) : Enter which piece to be moved (0 to Quit) & new Position

(User) : 0 0

(Comp) : Bye, Thank you !!

**Q.2** Write a Java program to create a Person class that contains name and age as member fields. This class implements a ‘Comparable’ interface that will allow us to sort the object of class Person. Create two objects of the person class and compare their age. Finally print a message indicating if they are of the same age or who is elder among the two.

**Q.3** Define a class Show which contains private members name (type String), cost (type double). This class implements the comparable interface as shown below.

class Show implements Comparable<Show>

{ private String name,

private double cost;

// fill in the constructor.

// fill in the implementation to compare by name

// Any other methods required..

} // of class Show

Define another class FindShow, having a method find(), as shown below.

class FindShow {

public static boolean find (Object[] arr, Object item) {

// provide the implementation to return true or false,

// depending on whether the show is present in arr or not.

}

}

Define a class containing the main method, which creates an array of objects of class Show and uses the class FindShow to search of a specific Show (by name). Finally display the Show details if present.

Also implement the Comparator interface and sort the Shows on descending order of price and display.