

Name Archit Agrawal
Student ID: 202051213
Sign: Archit Agrawal



Date 10/11/21
Page 1

CS201: Object Oriented Design and Programming

Remote Midsem Exam

Q.1.

- 1 → class Name
- 2 → attributes of class 'Order'
- 3 → methods of class 'Order'
- 4 → Abstract Class
- 5 → Inheritance of Abstract Class (Generalisation)
- 6 → Aggregation
- 7 → Multiplicity
- 8 → Role Name
- 9 → One to Many Association
- 10 → Association
- 11 → public
- 12 → protected



3.

Interface

- Interface can have only abstract methods.
- Variables declared in a Java interface are by default final and static.
- An interface can extend another Java interface only.

Abstract class

- An abstract class can have abstract and non-abstract methods.
- An abstract class can have final, non-final, static, non-static variables.
- An abstract class can extend another Java class and implement multiple Java interfaces

4.

// Return the title of this video tape

```
public String getTitle() {
```

```
    return this.title;
```

```
}
```

// Return the classification of this video tape

```
public String getClassification() {
```

```
    return this.classification;
```

```
}
```

// Return the time of this video tape

// as a string in the format : 2:06

```
public String getTime() {
```

```
    int hours = this.time / 60;
```

```
    int minutes = this.time % 60;
```

```
    String h = Integer.toString(hours);
```

```
    String m = Integer.toString(minutes);
```

```
    String s = "";
```

```
    s = h + ":" + m;
```

```
s = hours + ":" + minutes;
```

```
    return s;
```

```
}
```


// set a new classification for this video tape

```
public void setClassification(String s) {
```

```
    this.classification = s;
```

```
}
```

// Print the details of video tape to output

// terminal in the format

| Arman Singh (COMEDY) 2:16

```
public String showDetails() {
```

```
    String s = this.title + "(" + this.classification + ")";
```

```
    s += this.getTime();
```

```
    s += this.getTime();
```

```
    return s;
```

```
}
```

```
public void showDetails() {
```

```
    String s = this.title + "(" + this.classification + ")";
```

```
    s += this.getTime();
```

```
    System.out.println(s);
```

```
}
```

Name: Anshul Agrawal

Student ID: 202051213

Sign: Anshul Agrawal



5.

```
class ISBN {
```

```
    String
```

```
    protected isbnNo;
```

```
    // constructor
```

```
    public ISBN (String s) { this.isbnNo = s; }
```

```
    // getter - setter
```

```
    public String getISBN() { return this.isbnNo; }
```

```
    public void setISBN (String s) {
```

```
        this.isbnNo = s;
```

```
    }
```

```
}
```

```
class Book {
```

```
    private String name;
```

```
    private String author;
```

```
    private String publisher;
```

```
    private String city;
```

```
    private String date;
```

```
    private float price;
```

```
    private String isbnNum;
```

```
    // constructor
```

```
    public Book (String name, String author, float price)
```

Name: Archit Agrawal

Student ID: 202051213

Sign: Archit Agrawal



Page 6

```
this.name = name;  
this.author = author;  
this.price = price;
```

```
}
```

```
public void setBookISBN(String s) { this.isbnNum = s; }
```

```
public String getAuthor() { return this.author; }
```

```
public String getBookISBN() { return this.isbnNum; }
```

```
public void printDetails() {
```

```
    System.out.println("Book Title: " + this.name);
```

```
    System.out.println("Book Author: " + this.author);
```

```
    System.out.println("Book Publisher: " + this.publisher);
```

```
    System.out.println("ISBN: " + this.isbnNum);
```

```
}
```

```
}
```


Name: Archit Agrawal

Student ID: 202051213

Sign: 



6.

The 'name' and 'id' defined in the 'Employee' class have private access. Hence, they cannot be inherited by 'Manager' class.

The following changes are required:

- ★ In line [3], change 'private' to 'protected'
- ★ In line [4], change 'private' to 'protected'

Now, that the 'name' and 'id' ~~are~~ have protected access, they become private in the 'Manager' class. The 'name' and 'id' of 'Manager' object can be accessed using the 'getName()' and 'getID()' method defined in super class as these methods have public access.

- ★ In line [22], change return type 'void' to 'String'.

As 'department' is of ~~return~~ String type, return type of the getDeptment() should be 'String'.