King Saud University

College of Computer and Information Sciences
Computer Science Department

Course Code:	CSC 113
Course Title:	Computer Programming II
Semester:	Fall 2022
Exercises Cover Sheet:	**Midterm Exam**

Student Name:	
Student ID:	
Student Section No/ Time.	

Tick the Relevant	Computer Science B.Sc. Program ABET Student Outcomes	Question No. Relevant Is Hyperlinked	Covering%
	a) Apply knowledge of computing and mathematics appropriate to the computer science;		
	b) Analyze a problem, and identify and define the computing requirements appropriate to its solution		
	c) Design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;		
	d) Function effectively on teams to accomplish a common goal;		
	e) Understanding of professional, ethical, legal, security, and social issues and responsibilities;		
	f) Communicate effectively with a range of audiences;		
	g) Analyze the local and global impact of computing on individuals, organizations and society;		
	h) Recognition of the need for, and an ability to engage in, continuing professional development;		
	i) Use current techniques, skills, and tools necessary for computing practices.		
	j) Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;		
	k) Apply design and development principles in the construction of software systems of varying complexity;		

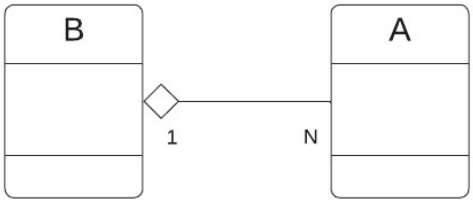
Name :_____ **ID:**_____

QUESTION 1:

Write the answers in the following table: **11 marks**

1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	

1- When the relationship between two classes as follows, then deleting B will delete _____ of A elements



A. all
B. some
C. one
D. none

2- Which of the following keywords can be used in a subclass to call the constructor of the parent class?

A. super
B. this
C. parent
D. extend

Name :_____ **ID:**_____

3- What is the output of the below Java program (assume no compilation/syntax errors)?

```
class Game
{ int runs; }

class Testing
{
  public static void main(String[] args)
  {
    Game g1 = new Game();
    g1.runs = 250;
    Game g2;
    g2 = g1;
    g2.runs = 300;
    System.out.println("Runs= " + g1.runs);
  }
}
```

A. Runs= 0
B. Runs= 250
C. Runs= 300
D. Error

4- In a Multi-Level Inheritance in Java, the last subclass inherits methods and properties of ____.
A. Only one immediate Superclass
B. Two classes above it.
C. All classes above it.
D. None

5- Subclass always has access to the superclass attributes and methods
A. True
B. False

6- Choose a correct statement about Java Interfaces.
A. Interface contains only abstract methods by default
B. A Java class can implement multiple interfaces
C. An Interface can extend or inherit another Interface
D. All the above

Name :_____ **ID:**_____

7- Which is the missing code to successfully compile the below Java program with abstract classes and Interfaces?

```
public interface A
{  void a( ); }
public abstract class B implements A
{  abstract void b ( ); }

public class C extends B
{
  // Missing methods
}
```

A	B	C	D

`@Override`
`public void a()`
`{ }`
`@Override`
`void b() {}`
	`@Override`
	`public void a()`
	`{ }`
		`@Override`
		`void b() {}`
			`@Override`
			`public void`
			`A.a() { }`
			`@Override`
			`void B.b() {}`

8- What will be the output of the following Java program (assume no compilation/syntax errors)?

```
public static void main(String
args[]) {
    int sum =  summer(4);
    System.out.println(sum);
          }
```

```
public static int summer(int in)
        {
          int sum = 0;
          if (in ==1)
              return 1;
          sum = in + summer(--in);

          return sum;
        }
```

A. 10
B. 6
C. 0
D. Infinite loop

9- What will be the output of the following Java program (assume no compilation/syntax errors)?

```
public static void main(String args[])
{
          int sum = multiplier(4);
              System.out.println(sum);
          }
```

```
static int multiplier(int in) {
              int mul = 0;
              if (in == 1) {
                  return 1;
              }
              mul = in * multiplier(in - 2);
              return mul;
          }
```

Name :_____ **ID:**_____

A. 24
B. 8
C. 0
D. Infinite loop

10- Which of these is NOT a correct statement?

A. Every class containing abstract method must be declared abstract
B. Abstract class defines only the structure of the class not its implementation
C. Abstract class can be initiated by `new` operator
D. Abstract class can be inherited

11- What is the error in the following Java program?

```
class A
    {
        private int i;
        protected int j;
    }
```

```
final class B extends A
    {
        private int j;
        void display()
        {
            super.j = 3;
            System.out.println(i + " " + j);
        }
    }
```

A. class B cannot inherit since it is `final`
B. `j` is defined in class A and B
C. `display` cannot access data member `i`
D. `display` cannot access data member `j`

Name :_____ **ID:**_____

Question 2: What is the output of main method (assume no compilation/syntax errors)?

```
public class Base {

    public Base() {
        System.out.println("Con Base 1");
    }

    public Base(String name) {
        System.out.println("Con Base 2");
    }

    public void m1() {
        System.out.println("Base m1");
    }

    public void m2() {
        System.out.println("Base m2");
    }

}
```

```
public class SubA extends Base {

    public SubA() {
        System.out.println("Con SubA");

    }

    public void m2() {

        System.out.println("SubA m2");
    }
}
```

```
public class SubB extends Base {

    public SubB() {
        super("Constructor");
        System.out.println("Con subB");
    }

    public void m1() {
       System.out.println("Sub B method 1");
    }
}
```

```
public class SubC extends SubB {

    public void m2() {
        System.out.println("Sub C method
2");
    }
}
```

```
class main {

    public static void main(String args[]) {
        Base[] array = {
            new SubB(),
            new SubC(),
        };
        for (int i = 0; i < array.length; i++) {
            array[i].m1();
            array[i].m2();

        }
    }
}
```

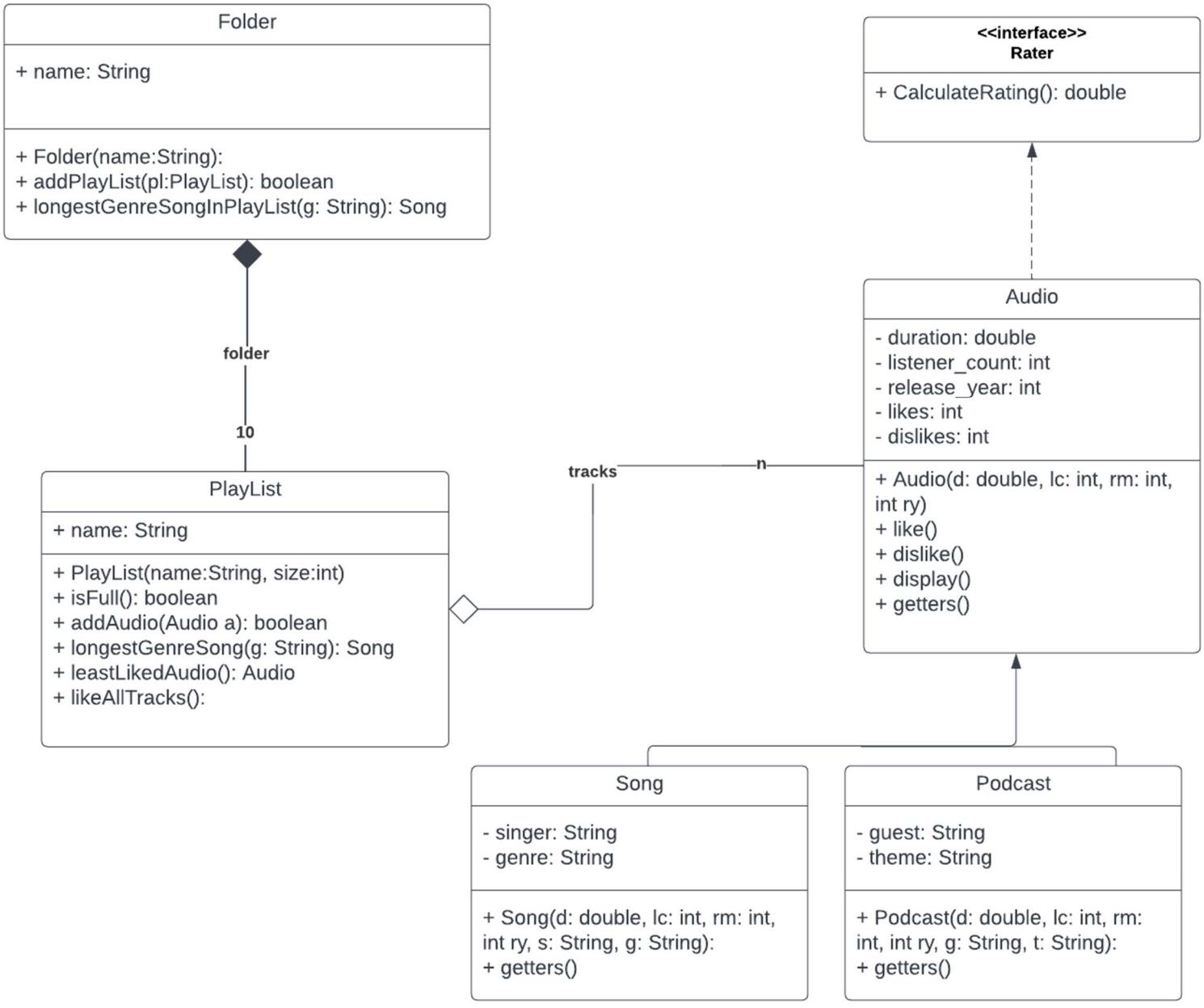
Name :_____ **ID:**_____

Answer: 4 marks

```
Con Base 2
Con subB
Con Base 2
Con subB
Sub B method 1
Base m2
Sub B method 1
Sub C method 2
```

Name :_____ **ID:**_____

Question 3: ***Consider the following UML class diagram:***



Name :_____ ID:_____

Interface Class ***Rater:***

- Methods:
  - ***CalculateRating():*** this method calculate the rating of the audio based on its type as follows:
    - ***Song*** returns the rating based on the following formula:
      - ***(likes/ (likes+ dislikes)) * 100***
    - ***Podcast*** returns the rating based on the following formula:
      - ***(likes/ (likes+ (dislikes*1.2))) * 100***

Class ***Audio:***

- Attributes:
  - **duration:** the duration of an audio in seconds.
  - **listener_count*:*** number of lisinter for the audio.
  - ***release_year:*** the year of release date.
  - ***likes:*** number of likes for the audio.
  - ***dislikes:*** number of dislikes for the audio.

- Methods:
  - ***Audio(d: double, lc: int, rm: int, int ry) : constructor***.
  - ***like():*** increase the number of likes by 1.
  - ***dislike():*** increase the number of dislikes by 1.
  - ***display():*** prints all the information of the object .
  - ***getters() :*** return the value of each attribute.

Class ***Song:***

- Attributes:
  - ***singer:*** the name of the singer.
  - ***genre***: category of the song
- Methods:
  - ***Song(d: double, lc: int, rm: int, int ry, s: String, g: String): constructor.***
  - ***getters() :*** return the value of each attribute.

Class ***Podcast:***

- Attributes:
  - ***guest:*** the name of the guest for the podcast.
  - ***theme:*** category of the podcast

Name :_____ ID:_____

- Methods:
  - ***Podcast(d: double, lc: int, rm: int, int ry, g: String, t: String): constructor***.
  - ***getters() :*** return the value of each attribute.

Class ***PlayList:***

- Attributes:
  - ***name:*** the name of the playlist.
  - ***size:*** the size of the playlist
- Methods:
  - ***PlayList(name:String, size:int): constructor***
  - ***isFull():*** return true if the playlist is full, and false otherwise.
  - ***addAudio(a: Audio):*** This method will add the audio ***a*** to the array. The method will return true if the audio is added successfully. Otherwise, it will return false.
  - ***longestGenreSong(g: String):*** This method will return the longest song in duration from genre **g**.
  - **leastLikedAudio():** This method will return the audio that has the minimum number of likes
  - ***LikeAllTracks():*** this method will like all the audio files in the playlist.

Class ***Folder***

- Attributes:
  - ***name:*** the name of the folder.
- Methods:
  - ***Folder* (*name:String*).**
  - **addPlaylist(pl:PlayList):** This method will the playlist pl in the array.
  - **longestGenreSongInPlayList(g: String):** This method returns the longest song from genre g in all playlists.

Name :_____ **ID:**_____

Answer the following questions: **2 marks**

A. Name the class(es) that will implement display() method?

Answer: Audio, song, podcast

B. Name the class(es) that will implement calcualteRating() method?

Answer: song, podcast

1- Complete the following methods.

```
public boolean addAudio(Audio a){ 2 marks

          if(isFull())
               return false;
          else{
               tracks[nb++] = a;
               return true
          }

}
public Audio leastLikedAudio() { 3 marks

   Audio min = tracks[0];
     for(int i=0; i < nb; i++)
          if(tracks[i].getlikes() < min.getlikes())
```

Name :_____ **ID:**_____

```
                min = tracks[i];


return min;



}
```

public Song longestGenreSong(String g) { **3 marks**

```
Song longest = null;
        for (int i = 0; i < nb; i++)
            if (tracks[i] instanceof Song && (Song)
tracks[i].getGenre().equlas(g)) {
                if (longest == null) {
                    Longest = (Song) tracks[i];
                } else if (longest.getDuration() <
tracks[i].getDuration()) {
                    Longest = (Song) tracks[i];
                }
}


return longest;
```

Name :___________________________________   ID:__________________

}