FACULTY OF COMPUTING
& INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY

FCIT
KAU

كلية الحاسبات
وتقنية المعلومات
جامعة الملك عبدالعزيز

Entertainment Management System(BuyTicket)

Assignment 4

CPCS203 Programming-II Spring 2019

Delivery Date: Sunday 31/3/2019

Instructions

- This program must ONLY be submitted on the Blackboard!
- This project worth 6% of the overall module marks (100%).
- NO assignment will be accepted after 11:59 pm for any reason
- Students can submit their assignment between 11 and 11:59 PM but in this case, it will be considered as late submission, and they will lose 2 points from the total mark of the assignment.
- For discussion schedule, check the captain name, date and time on the BlackBoard.
- Further information is provided in the course syllabus.
- ***Further information is provided in the course syllabus.***

Objectives

- Practice on the use of the **inheritance**, and **dynamic binding**.
- Learn to use and implement **polymorphism** and **object explicit casting**.
- Learn to use and implement **ArrayList** class.
- Learn to use the **instanceof** operator.
- Learn to use **abstract classes** and **interface**

How to submit your assignment?

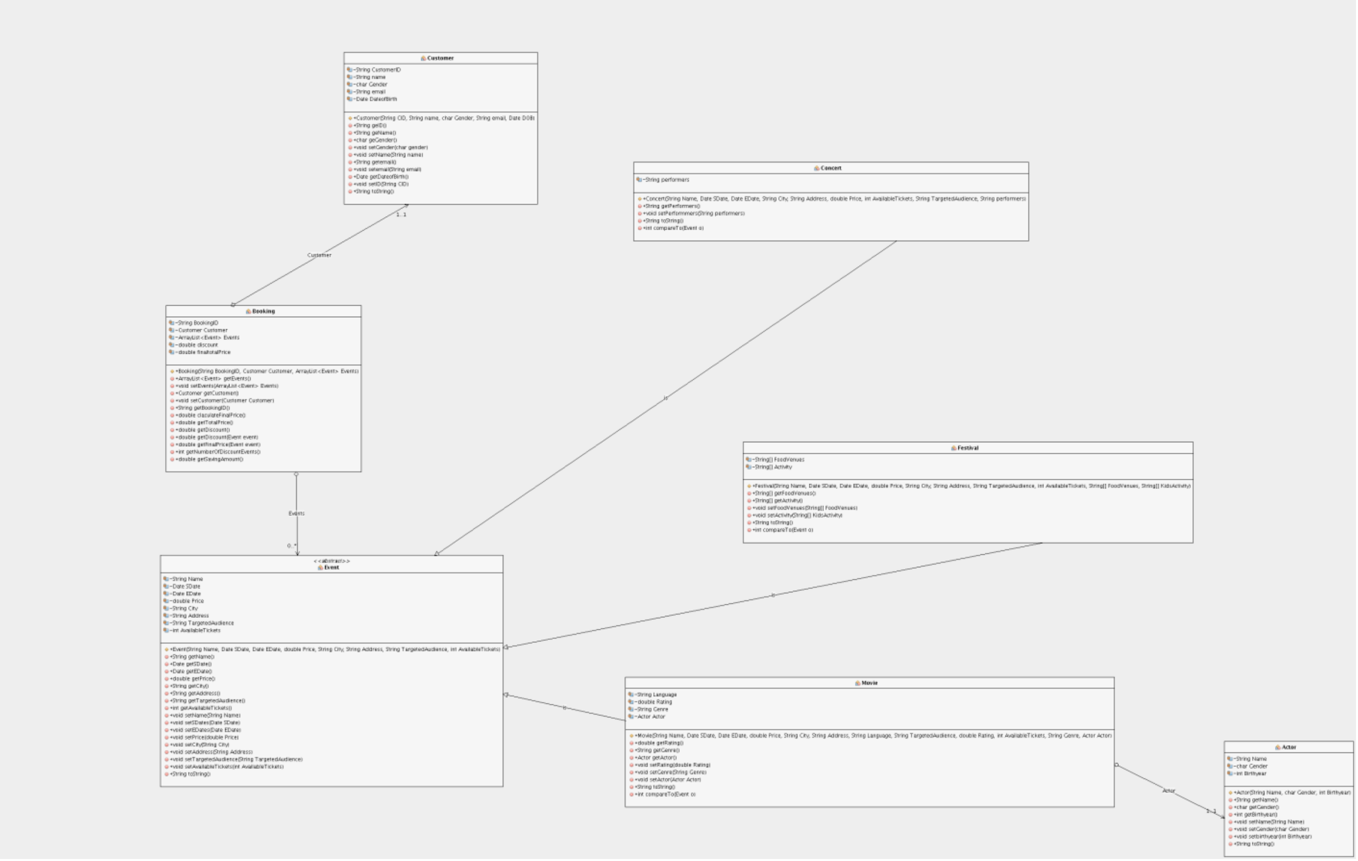
- Submit your assignment on the Blackboard ONLY.
- Make sure to add your names / IDs / Section / Your name / Assignment number at the beginning of your program

Files provided with assignment

- One input file: Input.txt
- One output file: Output.txt

The BuyTicket Software Description

BuyTicket is a software that helps users to buy a ticket for either movie, festival, or concert. The system allows customers to check the available events and buy a ticket for a different type of customers (adult or children). The system has three types of events: **movie**, **festival**, or **concert**. The event class is a super-class that has the common attributes and methods in all kinds of events (see the figure below). The three sub-classes **movie**, **festival**, and **concert** each extends the super-class and define other more specific attributes and methods. The system also allows customers to get a **special discount only on festival** according to their age. The following sections provide information about the classes and commands in the BuyTicket System. (Note: You might need to zoom in to visualize the diagram clearly. Note also that the diagram doesn't include the main class).


Customer
−String CustomerID
−String name
−char Gender
−String email
−Date DateofBirth
+Customer(String CID, String name, char Gender, String email, Date DOB)
+String getID()
+String geName()
+char geGender()
+void setGender(char gender)
+void setName(String name)
+String getemail()
+void setemail(String email)
+Date getDateofBirth()
+void setID(String CID)
+String toString()
1..1
Customer
Concert
−String performers
+Concert(String Name, Date SDate, Date EDate, String City, String Address, double Price, int AvailableTickets, String TargetedAudience, String performers)
+String getPerformers()
+void setPerformmers(String performers)
+String toString()
+int compareTo(Event o)
Booking
−String BookingID
−Customer Customer
−ArrayList<Event> Events
−double discount
−double finaltotalPrice
+Booking(String BookingID, Customer Customer, ArrayList<Event> Events)
+ArrayList<Event> getEvents()
+void setEvents(ArrayList<Event> Events)
+Customer getCustomer()
+void setCustomer(Customer Customer)
+String getBookingID()
+double claculateFinalPrice()
+double getTotalPrice()
+double getDiscount()
+double getDiscount(Event event)
+double getFinalPrice(Event event)
+int getNumberOfDiscountEvents()
+double getSavingAmount()
Festival
−String[] FoodVenues
−String[] Activity
+Festival(String Name, Date SDate, Date EDate, double Price, String City, String Address, String TargetedAudience, int AvailableTickets, String[] FoodVenues, String[] KidsActivity)
+String[] getFoodVenues()
+String[] getActivity()
+void setFoodVenues(String[] FoodVenues)
+void setActivity(String[] KidsActivity)
+String toString()
+int compareTo(Event o)
Events
0..*
<<abstract>>
Event
−String Name
−Date SDate
−Date EDate
−double Price
−String City
−String Address
−String TargetedAudience
−int AvailableTickets
+Event(String Name, Date SDate, Date EDate, double Price, String City, String Address, String TargetedAudience, int AvailableTickets)
+String getName()
+Date getSDate()
+Date getEDate()
+double getPrice()
+String getCity()
+String getAddress()
+String getTargetedAudience()
+int getAvailableTickets()
+void setName(String Name)
+void setSDates(Date SDate)
+void setEDates(Date EDate)
+void setPrice(double Price)
+void setCity(String City)
+void setAddress(String Address)
+void setTargetedAudience(String TargetedAudience)
+void setAvailableTickets(int AvailableTickets)
+String toString()
Movie
−String Language
−double Rating
−String Genre
−Actor Actor
+Movie(String Name, Date SDate, Date EDate, double Price, String City, String Address, String Language, String TargetedAudience, double Rating, int AvailableTickets, String Genre, Actor Actor)
+double getRating()
+String getGenre()
+Actor getActor()
+void setRating(double Rating)
+void setGenre(String Genre)
+void setActor(Actor Actor)
+String toString()
+int compareTo(Event o)
Actor
1..1
Actor
−String Name
−char Gender
−int Birthyear
+Actor(String Name, char Gender, int Birthyear)
+String getName()
+char getGender()
+int getBirthyear()
+void setName(String Name)
+void setGender(char Gender)
+void setbirthyear(int Birthyear)
+String toString()

Step 1: Creating The Described Classes

The first step in this project is **creating the required classes (six classes)** as shown on the UML diagram. You should create the classes with their methods, attributes, and constructors as shown the diagram. You will also create another class that performs that main methods in your program. **Note that** according the diagram above the class named **Event** is an **abstract parent** (**super class**) of three **concrete** classes **Movie**, **Festival**, and **Concert.** The class **Movie** has an object of class **Actor.** Class **Booking** that stores the booking information for a given customer has two objects one of classes **Customer** and **Event.** Class **Event** implements comparable interface.

Step 2: Adding the System Elements

Before allowing the customers to buy any tickets, the administrator needs to enter all the available movies, festivals, concerts, and all the customers. Four main commands need to be implemented in this step **AddMovie**, **AddFestival**, **AddConcert**, and **AddCustomer**. The following sections described each command in detail.

Command: AddMovie

This command is used to add a movie to the system with all its related information. To add a movie, the administrator needs to specify the following information: **Name, Starting Date, Ending Date, Price, City, Address, Language, Targeted Audience, Rating, Available Tickets, Genre, and Main Actor**. Note that The Main Actor is an object of **Actor** that has three main information **Main_Actor_Name**, **Main_Actor_Birth_Year**, and **Main_Actor_Gender** Check the following example and table.

Command Example
AddMovie--Bilal, 2/3/2019, 2/4/2019, 50.0, Jeddah, Red Sea Mall, English, Family, 3.8, 20, Animation, Adewale Akinnuoye, M, 1967

Field name	Type	Example
Name	String	Bilal
Starting Date	Date	2/3/2019
Ending Date	Date	2/4/2019
Price	Double	50.0
City	String	Jeddah
Address	String	Red Sea Mall
Language	String	English
Targeted Audience	String	Family
Rating	Double	3.8
Available Tickets	Int	20

Genre	String	Animation
Main_Actor_Name	String	Adewale akinnuoye
Main_Actor_Gender	Char	M
Main_Actor_Birth_Year	Int	1967

Command: AddFestival

AddFestival command is used to add all the information related to the available **festivals**. The information includes **Festival Name, Starting Date, Ending Date, Price, City, Address, Targeted Audience, Available Tickets, Food Venues, and Activities**. Check the following example and table. Note that **Food Venues** and **Activities** are both arrays of String.

Command Example
`AddFestival--Wild Desert, 27/2/2019, 10/3/2019, 25.0, Khobar, Khobar front sea, All, 600, FoodTruck cofeeshops, zoo clown`

Field name	Type	Example
Name	String	Wild Desert
Starting Date	Date	27/2/2019
Ending Date	Date	10/3/2019
Price	double	25.0
City	String	Kobar
Address	String	Kobar front sea
Targeted Audience	String	All
Available Tickets	Integer	600
Food Venues	String []	Burgerking MacDonald Frozenyogrt
Activities	String []	Zoo clown

Command: AddConcert

AddConcert command is used to **add all the information related to a concert event**. Concerts information include the **Concert_Name, Strating Date, Ending Date, Price, City, Address, Targeted_Audience , Available_Tickets, Performers_Names**. Check the following example and table.

Command Example
`AddConcert--Big Musical Festival, 23/9/2018, 24/9/2018, Jeddah, Jeddah Stadium, 0.0, 0, Male, 11 performers`

Field name	Type	Example
Name	String	`Big Musical Festival`
Starting Date	Date	`23/9/2018`
Ending Date	Date	`23/9/2018`
Price	double	`0.0`
City	String	`Jeddah`
Address	String	`King Abdullah Sprot City`
Target Audience	String	`Male`
Available Tickets	Integer []	`0`
Performers	String	`11 performers`

Command: AddCustomer

This command will add a customer to the system. To add a customer to the system needs **customer ID, customer name, address, and email**. Note that the **customer ID** will be generated by the program as a **unique 6-didgit counter started from 000001.**

Command Example
`AddCustomer--Ameerah Kareem, F, Jeddah, 12/2/1978, aa@gmail.com`

Field name	Type	Example
Customer ID	String	`000001`
Customer name	String	`Ameerah Kareem`
Gender	Char	`F`
City	String	`Jeddah`
Date of Birth	Date	`12/2/1978`
email	String	`aa@gmail.com`

Note that in this assignment the sizes of the arrays are not given so you **MUST** use **ArrayList** to store all the available events and another **ArrayList** to store the current customers.

Step 3: Buying a Ticket

After entering all the events along with all the customers. The system allows the users to ***BuyTicket*** for the registered customers to attend one of the available events.

Command: BuyTicket

This command is used to buy a ticket of an event either festival, movie, or/ and concert for any registered customer. Note that when booking an event for a customer the **available tickets will be reduced**, and a **special discount will be applied** according to the customer age. The following tables show the content of the BuyTicket command.

Command Example

```
BuyTicket--The mule, 7/3/2019
BuyTicket--Al Khamsa Legacy, 31/1/2019
BuyTicket--Wild Desert, 27/2/2019
For--Ahmad Ali
Submit
```

Field name	Type	Example
Event name	String	`The mule`
Booking Date	String	`7/3/2019`
Event name	String	`Al Khamsa Legacy`
Booking Date	String	`31/1/2019`
Event name	String	`Wild Desert`
Booking Date	String	`27/2/2019`
Customer name	String	`Ahmad Ali`

BuyTicket command will do the following:

- Search for the events and customers in the system. Print an error message if the event or the customer does not exist.
  For example, if the customer is not existed, the system will print *The customer Ali Ahmed is not found.*
  If the event is not existed, the system print *The event the help is not found.*
- If the event the customer existed in the system, the system checks the availability of the event in the requested date. If the event is not available in the requested date show an error message.
  If the event is not available in the requested date, the system will print *the requested event Wonder Awakened is not available in the selected date.*
- Then the system Buy a ticket for the customer by creating an **object of booking** for the given customer and selected events and reduce the events' available tickets.

- Print an invoice of the amount the customer need to pay.
- Note that the system applies a **special discount** on festival only according to the following rules:
    - If the customer age is **less than or equal 3 years older** at the present time, the customer ticket will be for **Free** (**discount rate=100%).**
    - If the customer age is **less than or equal 15 years older** at the present time, the customer receives a **discount rate = 50%.**
    - Customers **older than 15 years old do not receive any discount**.

Step 4: Print information

After the customer purchase all the requested items, the system automatically prints an invoice of the purchased events and payments. The invoice should have all the **purchased items**, **their original price**, the **special discount**, and **the final price**. The following table shows an example of a printed receipt and other example are provided in the ***output.txt*** file. Note that, you **MUST** use **polymorphism, casting, and instanceof** when you are ***Print_ invoice***:

The output of the printing the invoice:

```
------------------------- Invoice Details -------------------------
Booking# 000002039
Customer #: 000002
-------------------------------------------------------------------
Booking                  Price        Discount     Final_price
The mule                 50.00        0.0          50.00
Al Khamsa Legacy         400.00       0.0          400.00
Wild Desert              25.00        0.0          25.00
-------------------------------------------------------------------
Number of discount items: 0
-Total Price: 475.00
-Final Price:  475.00
-Saving Amount: 0.00
-------------------------------------------------------------------
```

1. Print Sorted Available Events

Command: PrintSortMovies

This command will first sort available movies by **their ratings** in a descending order and print the list of **movies**. **PrintSortMovies** command will print the following list of movies as follows:

The output of the **PrintSortMovies** command:

```
------------------------------------------- Movie Sorted By Rating -------------------------------------------
Movie name                Dates                Actor                     Ratings                   Available Tickets
--------------------------------------------------------------------------------------------------------
The Upside                06/04/3919-06/04/3919     Nicole Kidman          4.5                        40
The Lego Movie 2          09/04/3919-09/04/3919     Chris Pratt            4.0                        20
Bilal                     02/04/3919-02/04/3919     Adewale Akinnuoye      3.8                        20
Glass                     02/03/3919-02/03/3919     Samuel Jackson         3.4                        55
The mule                  07/04/3919-07/04/3919     Bradley Cooper         3.3                        15
Total Dhamal              06/04/3919-06/04/3919     Ajay Devgn             2.5                        25
 --------------------------------------------------------------------------------------------------------
```

Command: PrintSortFestivals

This command will first sort available festival by **their starting date** in an ascending order and print the list of Festivals. **PrintSortFestivals** will print the following list of festival in order as follows:

The output of the **PrintSortFestivals** command:

```
           ---------------------------------------- Festival Sorted By Starting Date -----------------------------------------------
           Festival Name         Dates                 FoodVenues                       Activities            Available Tickets
           -------------------------------------------------------------------------------------------------------------------------
           Happy Land            21/02/19-08/03/19     [Burger, Pizza]                  [theater, show]       400
           Wild Desert           27/02/19-10/03/19     [FoodTruck, CofeeShops]          [zoo, clown]          600
           Black Diamond         03/03/19-06/04/19     [CoffeeShops, HealthyFood]       [Sporting_events]     340
           Rayeg                 08/03/19-09/03/19     [FoodTruck, CofeeShops]          [zoo, clown]          300
           Jeddah Fair is coming 20/03/19-22/03/19     [Coffeeshops]                    [group_activities]    400
           Fashion show          20/03/19-22/03/19     [Burger, pizza, CoffeeShops]     [Bridal_show]         340
-------------------------------------------------------------------------------------------------------------------------
```

Note that you **MUST** use the **interface comparable** and override the **compareTo** method to sort and print the **array** of the movies or festival using the **sort** method.

Important Notes:

- Use of abstract classes, Interface, polymorphism is mandatory.
- Use of Files, Reading/Writing from/on files
- Your program output must be exactly same as given sample output files.
- Your display should be in a readable form.
- Organize your code in separated methods.
- Document your code with comments.
- Use meaningful variables.
- Use dash lines between each method.
- Delayed submission will not be accepted and there will not be any extension of the project.

Deliverables:

- You should submit one zip file containing all java codes: BA1587412P4_BuyTickets.java where BA is your section, 1587412 your ID and A4 is program 4.
- **NOTE: your name, ID, and section number should be included as comments in all files!**

Input and Output Format

Your program must generate output in a similar format to the sample run provided. **Sample input:** See sample input file.
Sample output: See sample output files.