VAAL UNIVERSITY OF TECHNOLOGY
Inspiring thought. Shaping talent.

FACULTY OF APPLIED & COMPUTER SCIENCES

DEPARTMENT OF COMPUTER SCIENCES

DIPLOMA: INFORMATION AND COMMUNICATION TECHNOLOGY

SUBJECT	:	DEVELOPMENT SOFTWARE 3.1
SUBJECT CODE	:	ASDSX3A
FORMATIVE ASSESSMENT	:	
DATE	:	5 APRIL 2025
DURATION OF TEST	:	
EXAMINER(S)	:	MRS S MOYO
MODERATOR(S)	:	MR N SOGANILE
REQUIREMENTS	**:**	**NetBeans version 8.2 and above**

INSTRUCTIONS:

1. Students are not allowed to handle cell phones in the examination room and cell phone accessories, but not limited to earpieces, are not allowed.
2. Students are not allowed to have any electronic or memory devices in their possession in the examination room.
3. Students must put their bags in front of the venue.
4. **Answer ALL the questions**
5. Name your project as follows**: YourStudentNumber_Assesment1_Opp1**
6. Submit your zipped NetBeans project on VUTela via the link: **Assesment1_Opp1_submission link.**

MARKS:	**Total**	**=**	**100**
	Full marks	**=**	**100**

THE QUESTION PAPER CONISTS OF: 8 typed pages

Problem Description

SANCGA (The South African National Computer Gamers Association) is hosting a gaming competition. Both professional as well as amateur gamers can register to take part in the competition. A professional gamer is defined as a person who is a member of SANCGA and is ranked on the professional gamers list. On registration the gamers need to provide their **ID-number** (a six-digit string), **name**, **age** and **status**.

Status is either amateur or professional. Amateur gamers are divided into three categories, viz.,learner (school), student (university or college) and adult. Professional gamers also need to provide their rank (an integer value between 0 and 99) and the number of years that they have been registered with SANCGA.

A registration code must be generated for each gamer.

The registration code for an amateur gamer consists of the letter **A,** followed by the gamer's ID-number, a **dash** character (**-**) and a digit indicating the category of the gamer. The category codes are **1** for a learner, **2** for a student and **3** for an adult.

The registration code for a professional gamer consists of the letter **P**, followed by a **4-digit** number, a **dash** character and the **number of years** that the gamer has been registered with SANCGA (2 digits). The 4-digit number consists of the gamer's rank (2 digits) and age (2 digits).

All numbers need to have leading zeros if their length is shorter than the specified length.

Examples of how the registration codes are generated can be seen in the **example data and output** given below. Each concrete class must generate the registration code for its instances using a method named **genCode()**.

The registration fee for professional gamers is R500 with a discount of 5% for each year that the gamer has been registered with SANCGA, up to a maximum of 50%. Therefore, professional gamers that have been registered with SANCGA for longer than 10 years will get a discount of R250.

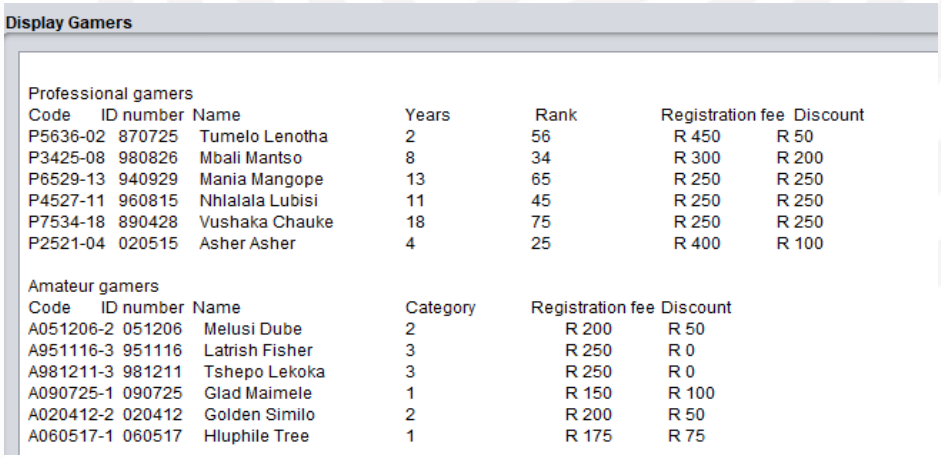
The registration fee for amateur gamers is R250. Students get a 20% discount, learners aged 16 and above get 30% discount and learners younger than 16 get 40% discount. Each of the concrete classes must use a method named **calculateDiscount(int fee)** to **calculate and return** the discount(as an integer) for a gamer. The method receives the registration fee via a parameter and calculate the discount if the fee supplied is not negative.

Example data

Amateur gamers			
ID-number	Name	Age	Category
051206	Melusi Duber	18	2
951116	Latricia Fisher	28	3
981211	Tshepo Lekoka	25	3
090725	Glad Maimele	14	1
020412	Golden Similo	21	2
060517	Hluphile Tree	17	1

Professional gamers				
ID-number	Name	Age	Registered years	Rank
870725	Tumelo Lenotha	36	2	56
980826	Mbali Mantso	25	8	34
940929	Mania Mangope	29	13	65
960815	Nhlalala Lubisi	27	11	45
890428	Vushaka Chauke	34	18	75
020515	Asher Asher	21	4	25

Example Output



QUESTION 1: Inheritance and polymorphism [34 Marks]

From the given information, identify all necessary classes with their attributes, constructors and methods. Two concrete classes, **AmGamer** and **ProGamer** and one abstract class, **Gamer**, should be amongst the identified classes. You should also create at least one interface, of your own, that should be implemented by your abstract class. Also use the information in the **example output** as a guide in your design process**. Produce code for all the identified classes and interface(s)**

Abstract methods must be used to ensure re-use of as much code as possible and to ensure polymorphic behavior. Therefore, no code or instance variables may be duplicated.

- **Attributes and constructor for the Gamer class**:

```
//declaring attribute
 private String id, name, code;
 private int age;
//default constructor
 public Gamer(){ }
//parameterized constructor
    public Gamer(String id, String name, int age)
        setId(id);
        setName(name);
        setAge(age);
    }
```

- It is imperative for the **setCode** setter to be given a public access modifier
- Refer to the example output when coding the **toString()** method for the 3 classes. NONE of the classes should display the registration fee and discount in its toString() method, these will be displayed by the GUI. Also note that the column headers in the example output are not displayed by the toString methods.

QUESTION 2: JUnit Testing [22 Marks]

Using the JUnit testing testing framework:

a. Verify if your **genCode** method for the **AmGamer** class generates the registration codes correctly.The correct codes for each gamer are included in the **example output**, use them as expected values. Create 3 test methods: **testGenCodeAmGamer1(),testGenCodeAmGamer2()** and **testGenCodeAmGamer3()** to test if your method correctly generates the codes for **Tshepo Lekoka, Glad Maimele** and **Golden Similo** respectively.

b. Verify if your genCode method for the ProGamer class generates the registration code correctly. Create 2 test methods: **testGenCodeProGamer1(), testGenCodeProGamer2()** to test your method for **Tumelo Lenotha and Vushaka Chauke** respectively

c. Verify if your **calculateDiscount** method is calculating the discounts correctly. The correct discounts for the different gamers are shown in the example output under the discount column. Use them as expected values.Create the methods below and check the correctness of your method using the data of the indicated gamer

testcalculateDiscountProGamer1() // Mbali Mantso
testcalculateDiscountProGamer2 //Mania Mangope
testcalculateDiscountAmGamer1() //Latricia Fisher
testcalculateDiscountAmGamer2() //Glad Maimela
testcalculateDiscountAmGamer3() //Hluphile Tree
testcalculateDiscountAmGamer4() //Golden Similo

QUESTION 3: GUI Application [44 Marks]

The application consists of only one form (JFrame **form**) shown in **Figure1**

Create the GUI using JFrame form populating it with the following input controls from the palette window:

-3 textFields: **txtID, txtName and txtRegisteredYrs** for ID number, name and registered years input entry respectively.

-3 combo boxes: **cmboAge**, **cmboRank**, **cmboCategory**.

-2 radio buttons: **radAmateur** and **radprofessional.** Group them and set radAmateur as a default radio button

-2 buttons: **btnRegisterGamer** and **btnViewAll**

-taGamer, a text area for displaying output.

-Replace the text **YourStudentNumber** on the form's title with your student number

(14)

- Populate cmboCategory with integers 1,2 and 3
- Populate cmboAge with integers from 10 to 80 and cmboRank with integers from 0 to 99 (use private methods to populate these 2 combo boxes).
- When radAmateur is selected, txtRegisteredYrs and cmboRank must be disabled as in Figure 1 and when radProfessional is selected, cmboCategory must be disabled. Write the code to do this.

(9)

- Declare an array list named **gamerArrList** that will be able to store both AmGamer and ProGamer objects.
- Declare the registration fees.

(4)

- Code **btnRegisterGamerActionPerformed(java.awt.event.ActionEvent evt)** procedure as follows:
    - Write code to check if the user has entered input in txtID, txtName and txtRegisteredYrs and display an error message if not, otherwise processing should continue.
    - Write code to read data from the controls and create objects of either AmGamer or ProGamer class depending on which radio button is selected.
    - Store the created object in the gamerArrList.

- Display the object on the text area taGamer as in **Figure 2** or **Figure 3**.
- clear the text fields for next input.
- Be sure to catch exceptions that may occur when the code is being processed.

(12)

- Code **btnViewAllActionPerformed(java.awt.event.ActionEvent evt)** as follows:
  - Write code to display ALL the data stored in gamerArrList as in **Figure 4**. (5)

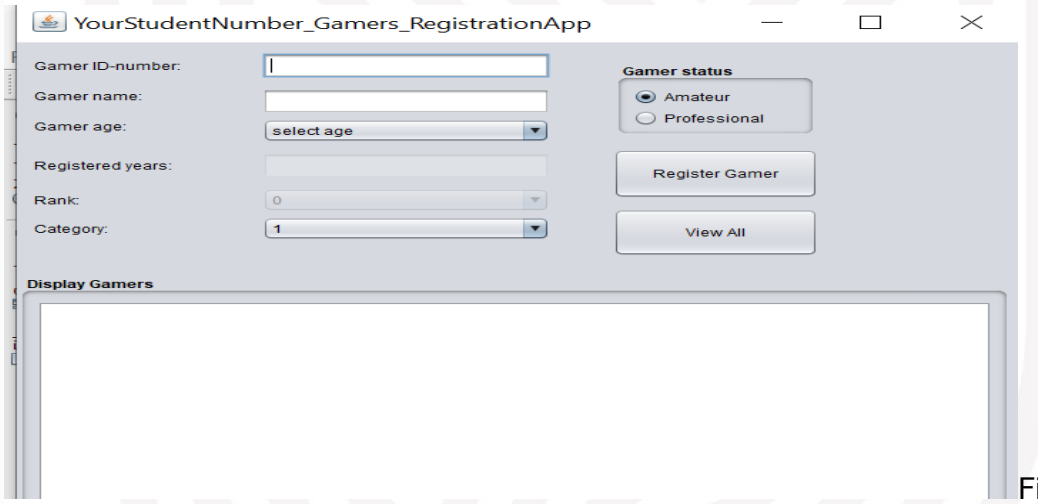


Figure 1

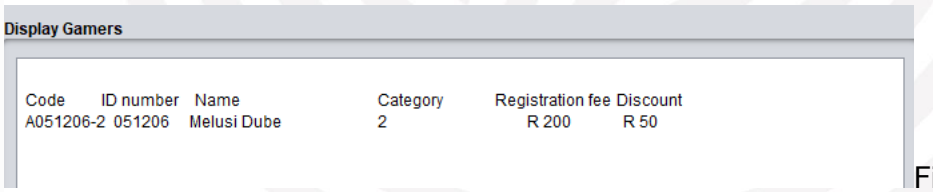


Figure 2

Figure 3

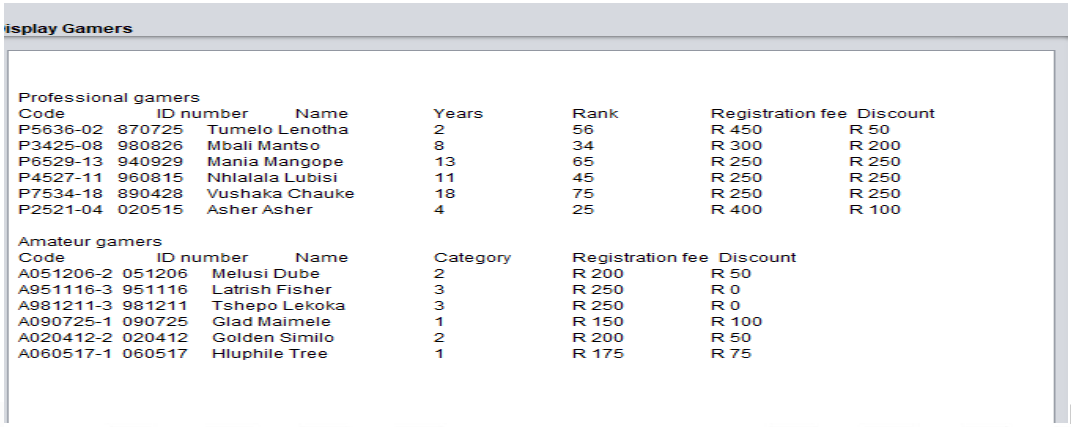


Figure 4

*******The End ****Good Luck*********