NAMIBIA UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Faculty of Computing and Informatics

Department of Computer Science

QUALIFICATION: Bachelor of Computer Science	
QUALIFICATION CODE: 07BACS	**LEVEL:** 5
COURSE: Object Oriented Programming	**COURSE CODE:** OOP521S
DATE: January 2020	**SESSION:** 2
DURATION: 3 HOURS	**MARKS:** 100

SUPPLEMENTARY/SECOND OPPORTUNITY EXAMINATION QUESTION PAPER	
EXAMINER	MR SIMON H. MUCHINENYIKA MR HERMAN KANDJIMI MRS ROSETHA KAYS MRS NDINELAGO NASHANDI MR STEVEN TJIRASO
MODERATOR:	MR COLIN STANLEY

THIS EXAM PAPER CONSISTS OF 2 PAGES
(Excluding this front page)

INSTRUCTIONS

1. This is a closed book examination with two sections; A and B.
2. Answer ALL questions in a separate writing booklet provided to you.
3. Total marks per section are indicated in **[]**, and () per question.
4. For Section A, answer each question on a new page.
5. NUST's examination rules and regulations apply.

SECTION A: [70 marks]

Cleary explain and distinguish the following terms from one another. Give example code in ALL cases.

1. primitive types and reference types (10 marks)

2. constructor overload and implicit constructor invocation (10 marks)

3. arrays and arraylists (10 marks)

4. default methods and abstract methods (10 marks)

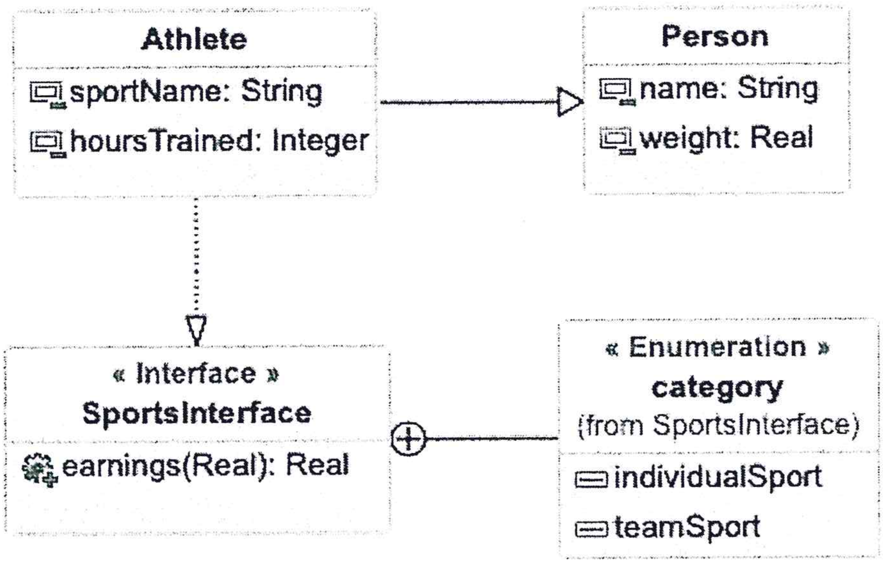
5. is-a and has-a (10 marks)

6. throws clause and throw statement (10 marks)

7. composition and aggregation (10 marks)

SECTION B: **[30marks]**

Kindly study the class diagram below and write code to fulfil the requirements given underneath.



1. Create the classes **Athlete** and **Person** as given in the class diagram. Add at least two constructors for each class, including a full constructor in each. Take note that all fields given as Real in the class diagram represent the type double. (5 marks)

2. Define **SportsInterface** that has an abstract method *weeklyEarnings()*, which returns a double and receives one parameter of type double. Also add an enum *category* as shown in the class diagram above. (4 marks)

3. Implement the **SportsInterface** assuming that *earnings* are proportional to the *hoursTrained* daily. This means you get *weeklyEarnings* by multiplying the *hoursTrained* by an athlete with the daily *wageRate* of the sport as given in the table below: (5marks)

sportName	wageRate
tennis	1700
soccer	2100
rugby	1200

4. Assuming that *weeklyEarnings* should not exceed 100 thousand weekly for each athlete, add a custom exception, **AthleteException** that will be thrown in the event that this condition is not met. Add all necessary code to handle this exception. (7marks)

5. In the driver class do the following operations:
   a. Create at least two **Athlete** objects using different constructors in (1). (2 marks)
   b. Demonstrate polymorphism and write an appropriate comment. (2 marks)
   c. Display all the properties of the objects you created to the screen. (2 marks)

6. Comments, readability and use of conventions. (3marks)

[END]

NAMIBIA UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Faculty of Computing and Informatics

Department of Computer Science

QUALIFICATION: Bachelor of Computer Science	
QUALIFICATION CODE: 07BACS	**LEVEL:** 5
COURSE: Object Oriented Programming	**COURSE CODE:** OOP521S
DATE: January 2020	**SESSION:** 2
DURATION: 3 HOURS	**MARKS:** 100

SUPPLEMENTARY/SECOND OPPORTUNITY EXAMINATION QUESTION PAPER	
EXAMINER	MR SIMON H. MUCHINENYIKA MR HERMAN KANDJIMI MRS ROSETHA KAYS MRS NDINELAGO NASHANDI MR STEVEN TJIRASO
MODERATOR:	MR COLIN STANLEY

THIS EXAM PAPER CONSISTS OF 2 PAGES
(Excluding this front page)

INSTRUCTIONS

1. This is a closed book examination with two sections; A and B.
2. Answer ALL questions in a separate writing booklet provided to you.
3. Total marks per section are indicated in **[]**, and () per question.
4. For Section A, answer each question on a new page.
5. NUST's examination rules and regulations apply.

SECTION A: [70 marks]

Cleary explain and distinguish the following terms from one another. Give example code in ALL cases.

1. primitive types and reference types (10 marks)

2. constructor overload and implicit constructor invocation (10 marks)

3. arrays and arraylists (10 marks)

4. default methods and abstract methods (10 marks)

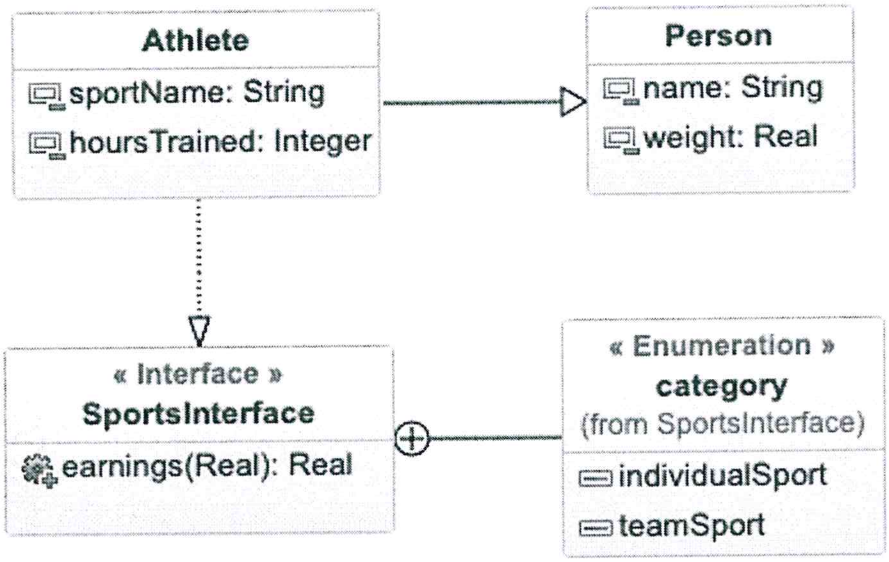
5. is-a and has-a (10 marks)

6. throws clause and throw statement (10 marks)

7. composition and aggregation (10 marks)

SECTION B: [30marks]

Kindly study the class diagram below and write code to fulfil the requirements given underneath.



1. Create the classes **Athlete** and **Person** as given in the class diagram. Add at least two constructors for each class, including a full constructor in each. Take note that all fields given as Real in the class diagram represent the type double. (5 marks)

2. Define **SportsInterface** that has an abstract method *weeklyEarnings()*, which returns a double and receives one parameter of type double. Also add an enum *category* as shown in the class diagram above. (4 marks)

3. Implement the **SportsInterface** assuming that *earnings* are proportional to the *hoursTrained* daily. This means you get *weeklyEarnings* by multiplying the *hoursTrained* by an athlete with the daily *wageRate* of the sport as given in the table below: (5marks)

sportName	wageRate
tennis	1700
soccer	2100
rugby	1200

4. Assuming that *weeklyEarnings* should not exceed 100 thousand weekly for each athlete, add a custom exception, **AthleteException** that will be thrown in the event that this condition is not met. Add all necessary code to handle this exception. (7marks)

5. In the driver class do the following operations:
   a. Create at least two **Athlete** objects using different constructors in (1). (2 marks)
   b. Demonstrate polymorphism and write an appropriate comment. (2 marks)
   c. Display all the properties of the objects you created to the screen. (2 marks)

6. Comments, readability and use of conventions. (3 marks)

[END]