**NED UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

**BS PROGRAMME OF USMAN INSTITUTE OF TECHNOLOGY**

**FIRST YEAR (COMPUTER SCIENCE)**

**SUMMER SEMESTER EXAMINATION 2022**

**BATCH 2019-20**

**OBJECT ORIENTED PROGRAMMING CS 121**

Time: 3 Hours Max. Marks 12

Q1 (2 marks) (4 marks)

|  |
| --- |
| **Student** |
| - name: String  - testScore: int  - grade: String |
| Student (name: String)  getName(): String  setTestScore(int): void  computeGrade(): void  getGrade(): String |

|  |
| --- |
| **GraduateStudent** |
|  |
| GraduateStudent(name: String)  computeGrade(): void |

|  |
| --- |
| **UndergraduateStudent** |
|  |
| UndergraduateStudent(name: String)  computeGrade(): void |

For UML Class diagram represent classes for an Undergraduate and a Graduate student both being inherited from the Student class which is an abstract base class. Compute grade function in student class will be abstract function. For an undergraduate student, the grade would be ‘Pass’ if the test score is greater than or equals to 60, and ‘Fail’ otherwise. However, for a graduate student the grade would be ‘Pass’ if the test score is greater than or equal to 70, and ‘Fail’ otherwise

Write Python code for defining the classes

Q2 (2 marks) (4 marks)

Write code for the **StudentList** class as described below

Design a class named **StudentList** that contains:

■ A private int data field named count for the student count.

■ A private list field named studentlist for the students.

■ Add function that adds the student in student list.

■ Search function that searches the student in student list.

■ Set marks function that takes student id and marks as input, this function calls search function and set marks for student.

■ Compute grade function that calls compute grade function for all students.

Q3 (4 marks) (8 marks)

Write code for following

Create negative mark exception class and raise exception at appropriate place if negative marks are entered for student. This class will have private marks field.

Create student not found exception class and raise exception at appropriate place if student is not found during search. This class will have private student name.

Create student marks not set exception class and raise exception at appropriate place if compute grade is called for student whose marks are not set. This class will have student name as private field.

Create invalid type exception class and raise exception at appropriate place if any other class type is provided other than student class.

The system displays the main menu as follows, all above defined exceptions are handled in below menu and the program should not halt due to any exception.

1. Add Student

A new employee is added to the Student roster

User would be prompted to decide student type and provide relevant details

1. Set Student marks

User would be prompted to provide student ID and marks

If the record is available against the student ID, marks will be set against the id. Else, an appropriate message would be displayed. If the marks are negative then user is displayed appropriate message and user is taken back to beginning of menu.

1. Compute student grades

Grades for all students will be computed. If any student marks are not set then user is taken back to beginning of menu.

1. Display Student Roster

A complete record of all the students is displayed. The details to be included is the name, ID and the grade of each student

1. Exit the Application

Q4 (4 marks) (8 marks)

Redraw the class diagram in Q1 and add StudentList class as explained in Q2, also add exception classes created in Q3 in one single UML diagram.

VIVA

Q1) (1 mark)

What is the output of the following program

class Fan:

    count = 0

    def \_\_init\_\_(self):

        Fan.count += 1

f1 = Fan()

f2 = Fan()

print(f1.count)

print(f2.count)

Q2) (1 mark)

Identify errors in below program

from abc import ABC, abstractmethod

class Parent(ABC):

    @abstractmethod

    def func(self):

        pass

class Child(Parent):

    pass

p = Parent()

c = Child()

Q3) (2 marks)

What is the output of the following program

class P1:

    def \_\_init\_\_(self):

        print("P1")

class P2:

    def \_\_init\_\_(self):

        print("P2")

class C(P1, P2):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

c = C()

Q4) (2 marks)

Is the code written according to UML diagram?

|  |
| --- |
| **BMI** |
| - name: String  - age: int  - weight: float  - height: float |
| BMI(name: String, age: int, weight: float, height: float)  getName(): String  getBMI(): float  getStatus(): String |

class BMI:

    def \_\_init\_\_(self, name, age, weight, height):

        self.name = name

        self.age = age

        self.weight = weight

        self.height = height

    def getName(self):

        pass

    def getBMI(self):

        pass

    def getStatus(self):

        pass