2 Class and Object in Python with GIS Application Emphasis

2.1 Question (self-check)

2.1.1 Instructions to Correct the Code

Below is a Python code snippet that defines a `Student` class. This class represents student records with attributes such as name, age, grade, status (active, graduated, dropped out), and coordinates (x, y) of the students' houses. The class also includes methods to calculate the distance from the student's house to the school.

```
The provided code contains intentional errors for you to identify and
correct.
```

2.1.2 Your tasks are:

1. **Identify Errors:** Examine the given code carefully and identify lines that contain errors.
2. **Correct Errors:** Select the correct option that identifies the error or provides the correct form of the problematic line.
3. **Answer Questions:** Answer the multiple-choice questions based on the provided code and concepts related to object-oriented programming in Python.

2.1.3 Code with errors

```
import math

# Ask the user to enter the location of the school
school_x = float(input("Enter the x-coordinate of the school: "))
school_y = float(input("Enter the y-coordinate of the school: "))

class Student:
    def __init__(self, name, age, grade, status, house_x, house_y):
        self.name = name
        self.age = age
        self.grade = grade
        self.status = status
        self.house_x = house_x
        self.house_y = house_y

    def __repr__(self):
        return f"Student({self.name}, {self.age}, {self.grade},
{self.status}, {self.house_x}, {self.house_y})"

    def display_status(self):
        print(f"Student {self.name} is currently {self.status}")

    def distance_to_school(self):
X        distance = math.sqrt((self.house_x - school_x)  2 +
(self.house_y - school_y))
        return distance
```

```
# Ask the user to enter student information
name = input("Enter student's name: ")
age = int(input("Enter student's age: "))
grade = input("Enter student's grade: ")
status = input("Enter student's status: ")
house_x = float(input("Enter the x-coordinate of the student's house:
"))
house_y = float(input("Enter the y-coordinate of the student's house:
"))

# Incorrect instantiation (Error 1: Missing arguments)
student1 = Student(name, age, grade, status, house_x, house_y)

# Incorrect method call (Error 2: Method name typo)
print(student1.displaystatus())

# Calculate distance to school (Error 3: Incorrect method)
print(f"Distance to school: {student1.distance_to_school()} units")
```

2.1.4 Part 1: Multiple Choice Questions:

Question 1: What is the error in the method `distance_to_school`?

A. The formula for calculating distance is incorrect.
B. The `math` module is not imported.
C. The y-coordinate calculation is missing exponentiation.
D. The `school_x` and `school_y` arguments are swapped.

Question 2: What is the correct form of the method call to display the status of the student?

A. `print(student1.display_status())`
B. `print(student1.displayStatus())`
C. `print(student1.Display_Status())`
D. `print(student1.displaystatus())`

Question 3: What is the error in the method `distance_to_school` call?

A. The `distance_to_school` method is missing arguments.
B. The `school_x` and `school_y` arguments are swapped.
C. The `math` module is not imported.
D. The y-coordinate calculation is missing exponentiation.

Part 2: Multiple Choice Questions: general multiple-choice questions based on the comments and lines of code in the `Student` class example:

4. What is the purpose of the block `# Ask the user to enter student information` in the code?

A. To initialize the attributes of the school.
B. To prompt the user to input data for a new student.
C. To display the student's status.
D. To calculate the distance to the school.

5. What is the function of `print(f"Distance to school: {student1.distance_to_school()} units")` in the code?

A. To initialize the distance_to_school method.
B. To display the status of the student.
C. To print the calculated distance from the student's house to the school.
D. To prompt the user to input the location of the school.

6. Why is `import math` used in the code?
   A. To allow the use of mathematical operations and functions.
   B. To prompt the user to input data.
   C. To display the student's status.
   D. To initialize the `__init__` method.

7. What is the purpose of `float(input("Enter the x-coordinate of the school: "))` in the code?
   A. To initialize the student's grade.
   B. To display the student's status.
   C. To calculate the distance to the school.
   D. To prompt the user to input the x-coordinate of the school.

2.2 Create, correct, test, and discuss the Python code for the `Student` class.

Instructions:

1. Create the Python Code

- Using the provided code below, create a Python script named `student.py`.
- This code contains intentional errors for you to identify and correct.

Code with errors

```
import math

# Define the location of the school (users enter the location of the
school first)
school_x = 0  # Example value, user can change
school_y = 0  # Example value, user can change

class Student:
    def __init__(self, name, age, grade, status, house_x, house_y):
        self.name = name
        self.age = age
        self.grade = grade
        self.status = status
        self.house_x = house_x
        self.house_y = house_y

    def __repr__(self):
        return f"Student({self.name}, {self.age}, {self.grade},
{self.status}, {self.house_x}, {self.house_y})"

    def display_status(self):
        print(f"Student {self.name} is currently {self.status}")

    def distance_to_school(self):
        distance = math.sqrt((self.house_x - school_x)2 + (self.house_y
- school_y))
        return distance

# Ask the user to enter student information
name = input("Enter student's name: ")
age = int(input("Enter student's age: "))
```

```
grade = input("Enter student's grade: ")
status = input("Enter student's status: ")
house_x = float(input("Enter the x-coordinate of the student's house: 
"))
house_y = float(input("Enter the y-coordinate of the student's house: 
"))

# Incorrect instantiation (Error 1: Missing arguments)
student1 = Student(name, age, grade, status, house_x, house_y)

# Incorrect method call (Error 2: Method name typo)
print(student1.displaystatus())

# Calculate distance to school (Error 3: Incorrect method)
print(f"Distance to school: {student1.distance_to_school()} units")
```

2. Identify and Correct Errors:

- Review the provided code and identify any errors.
- Correct each identified error in the code.
- The key errors to correct include:
    - Correct the `distance_to_school` method calculation.
    - Correct the method name typo in `print(student1.displaystatus())`.

3. Test the Code:

- Run the corrected script to ensure it functions as expected.
- Verify that the script correctly calculates the distance from the student's house to the school and displays the student's status.

4. Apply it Using Imaginary Data:

- Use imaginary data for student records to test the corrected script.
- Example data to use:
    - Student Name: Alice
    - Age: 16
    - Grade: 10th Grade
    - Status: Active
    - House Coordinates: (5, 10)
    - School Coordinates: (0, 0)

5. Discuss with Group Members:

- Share your findings and discuss any challenges you faced while correcting the code with your group members.
- Discuss how the corrections improved the functionality of the script.
- Explore additional features or improvements that could be added to the script.

By following these steps, you will learn how to identify and correct errors in a Python script, test it with data, and collaborate with your group members to enhance your understanding of object-oriented programming in Python.