BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Quiz 4
Semester: Fall 2024

Duration: 20 minutes
Full Marks: 15

CSE 470: Software Engineering

Name: **ID:** **Section:**

1. Write down any **code smells** that may have caught your attention. [3]
2. **Refactor** the program as needed. [7]
3. Calculate the SIX for the given `PostgradStudentHandler Class`. [5]

```
class StudentUserHandler:

    def get_details(self, name, student_id, department, cgpa, date_of_admission, credits, scholarship_amount):
        student_name = name
        student_id = student_id
        student_department = department
        student_cgpa = cgpa
        student_guardian_name = ""
        student_address = ""

        course_count = 0
        while course_count < 4:
            print(f"CourseNo {course_count} is completed")
            course_count += 1

        lowest_cgpa_details = AllCGPA().get_lowest_cgpa()
        highest_cgpa_details = AllCGPA().get_highest_cgpa()

        if (self.cgpa >= 3.8 and self.credits_completed >= 120 and
            self.department in ["Engineering", "Science"] and
            self.extracurricular_participation >= 3):
            print(f"{self.name} is eligible for a full merit scholarship.")

        for course_no in range(course_count):
            print(f"CourseNo {course_no} is completed")

    def update_details(self):
        print("You need to setup your studentDetails")
        print("You need to setup your personalDetails")

class UndergradStudentHandler(StudentUserHandler):
    def __init__(self):
        super().__init__()
        self.student_guardian_name = ""
        self.student_address = ""
```

```
    def pub_grd(self):
        print("Here is your grade")

    def take_courses(self):
        print("please take at least 4 courses")

class PostgradStudentHandler(UndergradStudentHandler):

    def pub_grd(self):
        print("Here is your grade")

    def provide_postgrad_student_details(self):
        print("please take at least 4 courses")
```

(NMI)

(NMO)

(NMA)

$$SIX = \frac{NMO \times DIT}{NMO + NMA + NMI} \times 100$$

$$= \frac{1 \times 2}{1 + 1 + 4} \times 100$$

$$= \frac{2}{6} \times 100$$

$$= \frac{1}{3} \times 100$$

$$= 33.33\ \%$$

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Quiz 4
Semester: Fall 2024

Duration: 20 minutes
Full Marks: 15

CSE 470: Software Engineering

Name: **ID:** **Section:**

1. Write down any **code smells** that may have caught your attention. [3]
2. **Refactor** the program as needed. [7]
3. Calculate the **SIX** for the given `ManagerEmployeeHandler Class`. [5]

```
class EmployeeUserHandler:
    def get_details(self, name, employee_id, department, salary, date_of_joining, total_work_hours, bonus):
        employee_name = name
        employee_id = employee_id
        employee_department = department
        employee_salary = salary
        employee_address = ""
        employee_supervisor_name = ""

        lowest_salary_details = AllSalaries().get_lowest_salary()
        highest_salary_details = AllSalaries().get_highest_salary()

        for task_no in range(total_tasks):
            print(f"TaskNo {task_no} is completed")

        task_no = 0
        while task_no < total_tasks:
            print(f"TaskNo {task_no} is completed")
            task_no += 1

        if (self.role == "Manager" and self.department == "Sales" and self.experience_years >= 5 and
            self.performance_score > 8 and self.projects_completed >= 10):
            print(f"{self.name} is eligible for a promotion to Senior Manager.")

    def update_details(self):
        print("You need to set up your employee details.")
        print("You need to set up your personal details.")

class RegularEmployeeHandler(EmployeeUserHandler):

    def publish_performance_review(self):
        print("Here is your performance review.")

    def asn_tsks(self):
        print("Please take at least 5 tasks.")
```

```
class ManagerEmployeeHandler(RegularEmployeeHandler):

    def __init__(self):
        self.employee_supervisor_name = ""
        self.regular_employee_leave_days = ""

    def publish_performance_review(self):
        print("Here is your team performance review.")

    def provide_team_details(self):
        print("Please provide the team details.")
```

NMO (overridden the default)

(NMO)

(NMA)

$$SIX = \frac{NMO \times DIT}{NMO + NMA + NMI} \times 100$$

$$= \frac{2 \times 2}{2+1+3} \times 100$$

$$= \frac{4}{6} \times 100$$

$$= \frac{2}{3} \times 100$$

$$= 66.66\%$$