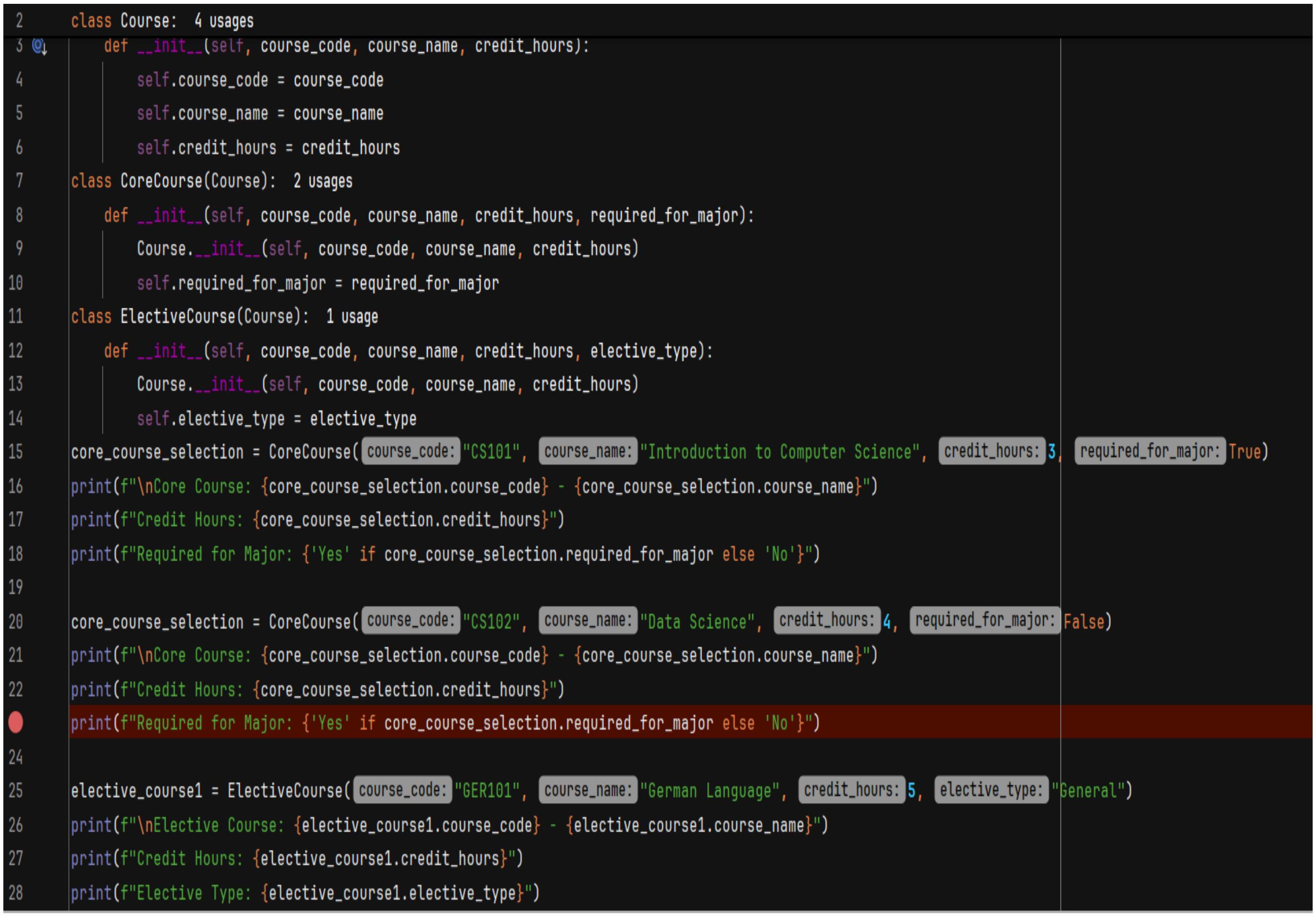
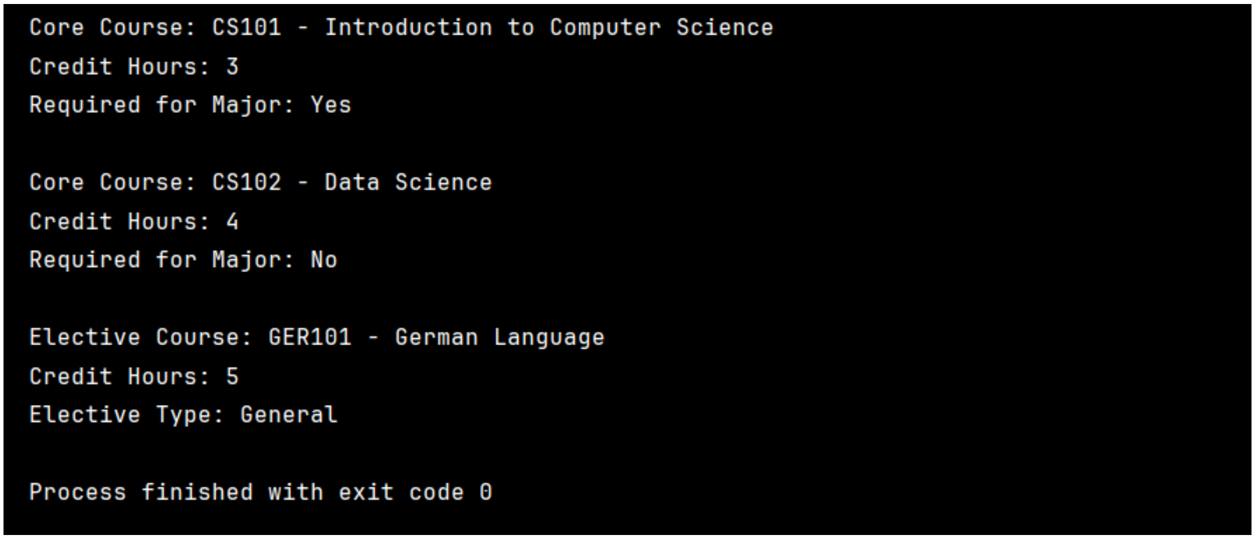
5_Python_Part_Assignment_OOPs

Question 1: (5 Marks) Build a program to manage a university's course catalog. You want to define a base class Course that has the following properties: course_code: a string representing the course code (e.g., "CS101") course_name: a string representing the course name (e.g., "Introduction to Computer Science") credit_hours: an integer representing the credit hours for the course (e.g., 3) You also want to define two subclasses CoreCourse and ElectiveCourse, which inherit from the Course class. CoreCourse should have an additional property required_for_major which is a boolean representing whether the course is required for a particular major. ElectiveCourse should have an additional property elective_type which is a string representing the type of elective (e.g., "general", "technical", "liberal arts").

Python_Assignment_5.py



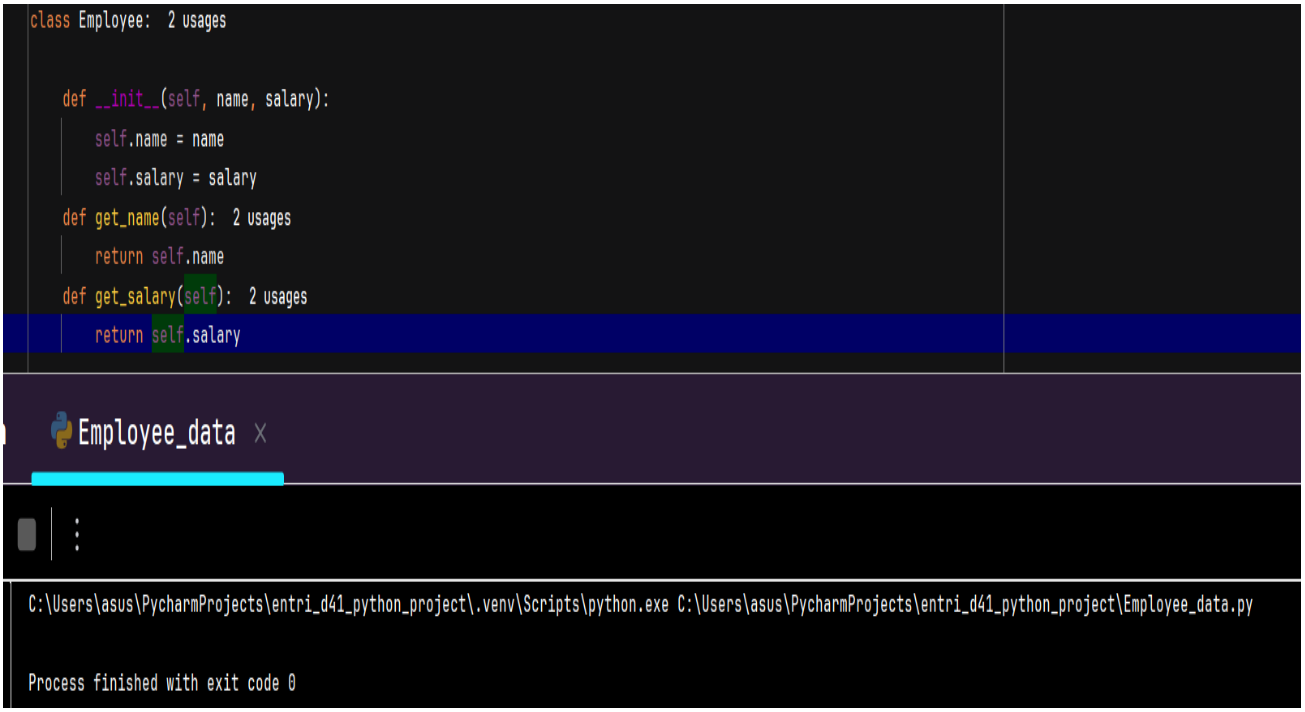
```
class Course:  4 usages
    def __init__(self, course_code, course_name, credit_hours):
        self.course_code = course_code
        self.course_name = course_name
        self.credit_hours = credit_hours
class CoreCourse(Course):  2 usages
    def __init__(self, course_code, course_name, credit_hours, required_for_major):
        Course.__init__(self, course_code, course_name, credit_hours)
        self.required_for_major = required_for_major
class ElectiveCourse(Course):  1 usage
    def __init__(self, course_code, course_name, credit_hours, elective_type):
        Course.__init__(self, course_code, course_name, credit_hours)
        self.elective_type = elective_type
core_course_selection = CoreCourse( course_code: "CS101",  course_name: "Introduction to Computer Science",  credit_hours: 3,  required_for_major: True)
print(f"\nCore Course: {core_course_selection.course_code} - {core_course_selection.course_name}")
print(f"Credit Hours: {core_course_selection.credit_hours}")
print(f"Required for Major: {'Yes' if core_course_selection.required_for_major else 'No'}")

core_course_selection = CoreCourse( course_code: "CS102",  course_name: "Data Science",  credit_hours: 4,  required_for_major: False)
print(f"\nCore Course: {core_course_selection.course_code} - {core_course_selection.course_name}")
print(f"Credit Hours: {core_course_selection.credit_hours}")
print(f"Required for Major: {'Yes' if core_course_selection.required_for_major else 'No'}")

elective_course1 = ElectiveCourse( course_code: "GER101",  course_name: "German Language",  credit_hours: 5,  elective_type: "General")
print(f"\nElective Course: {elective_course1.course_code} - {elective_course1.course_name}")
print(f"Credit Hours: {elective_course1.credit_hours}")
print(f"Elective Type: {elective_course1.elective_type}")
```



```
Core Course: CS101 - Introduction to Computer Science
Credit Hours: 3
Required for Major: Yes

Core Course: CS102 - Data Science
Credit Hours: 4
Required for Major: No

Elective Course: GER101 - German Language
Credit Hours: 5
Elective Type: General

Process finished with exit code 0
```

Question 2: (5 Marks) Create a Python module named employee that contains a class Employee with attributes name, salary and methods get_name() and get_salary(). Write a program to use this module to create an object of the Employee class and display its name and salary.

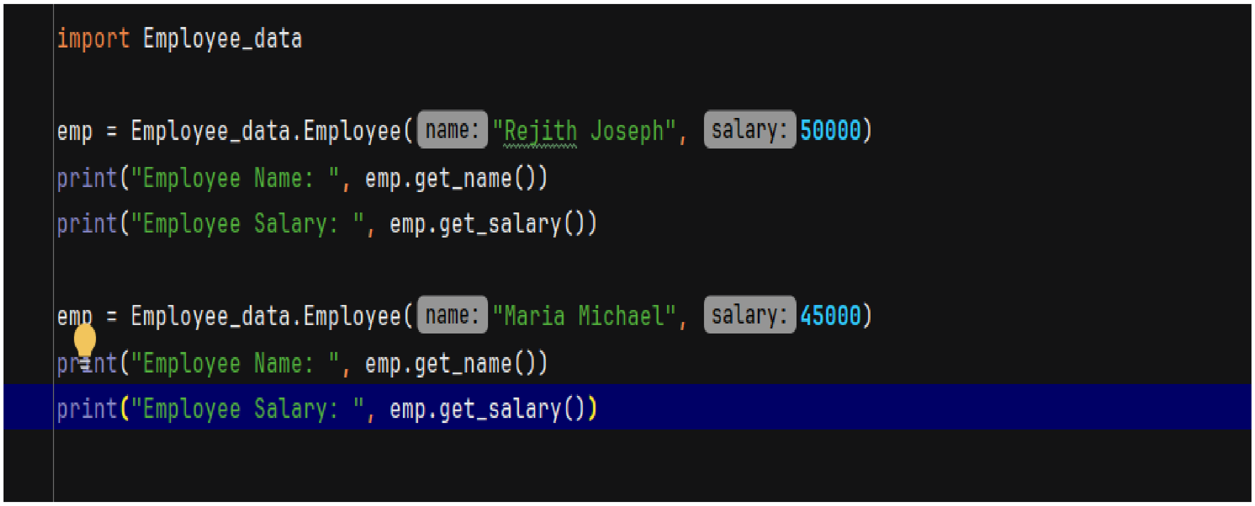
Employee_data.py



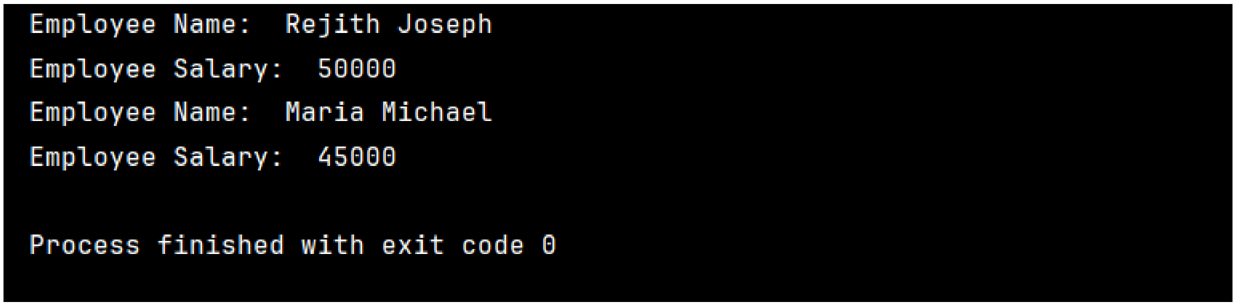
```
class Employee:  2 usages

    def __init__(self, name, salary):
        self.name = name
        self.salary = salary
    def get_name(self):  2 usages
        return self.name
    def get_salary(self):  2 usages
        return self.salary
```

```
C:\Users\asus\PycharmProjects\entri_d41_python_project\.venv\Scripts\python.exe C:\Users\asus\PycharmProjects\entri_d41_python_project\Employee_data.py

Process finished with exit code 0
```

Employee_With_Object.py



```
import Employee_data

emp = Employee_data.Employee(name: "Rejith Joseph", salary: 50000)
print("Employee Name: ", emp.get_name())
print("Employee Salary: ", emp.get_salary())

emp = Employee_data.Employee(name: "Maria Michael", salary: 45000)
print("Employee Name: ", emp.get_name())
print("Employee Salary: ", emp.get_salary())
```



```
Employee Name:  Rejith Joseph
Employee Salary:  50000
Employee Name:  Maria Michael
Employee Salary:  45000

Process finished with exit code 0
```