

```
entri_d41_python_project Version control Current File
in1.py Tuple.py functions.py filehandling.py Module.py test_module.py Oops.py Encapsulation.py Assignment6.py Assignment6(2).py
1 # Question 1: Build a program to manage a university's course catalog.You want to define a base class
2 # Course that has the following properties: course_code: a string representing the course code
3 # (e.g., "CS101") course_name: a string representing the course name (e.g., "Introduction to Computer Science")
4 # credit_hours: an integer representing the credit hours for the course (e.g., 3) You also want to
5 # define two subclasses CoreCourse and ElectiveCourse, which inherit from the Course class.
6 # CoreCourse should have an additional property required_for_major which is a boolean representing
7 # whether the course is required for a particular major. ElectiveCourse should have an additional property
8 # elective_type which is a string representing the type of elective (e.g., "general", "technical", "liberal arts").
9 class Course: 2 usages
10     def __init__(self, course_code, course_name, credit_hours):
11         self.course_code = course_code
12         self.course_name = course_name
13         self.credit_hours = credit_hours
14
15     def display(self): 2 usages
16         print(f"{self.course_code}: {self.course_name} ({self.credit_hours} Credit Hours)")
17 class CoreCourse(Course): 1 usage
18     def __init__(self, course_code, course_name, credit_hours, required_for_major):
19         super().__init__(course_code, course_name, credit_hours)
20         self.required_for_major = required_for_major
21
22     def display(self): 1 usage
```

```
entri_d41_python_project Version control Current File
in1.py Tuple.py functions.py filehandling.py Module.py test_module.py Oops.py Encapsulation.py Assignment6.py Assignment6(2).py

9 class Course: 2 usages
11     self.course_code = course_code
12     self.course_name = course_name
13     self.credit_hours = credit_hours
14
15 def display(self): 2 usages
16     print(f"{self.course_code}: {self.course_name} ({self.credit_hours} Credit Hours)")
17 class CoreCourse(Course): 1 usage
18     def __init__(self, course_code, course_name, credit_hours, required_for_major):
19         super().__init__(course_code, course_name, credit_hours)
20         self.required_for_major = required_for_major
21
22 def display(self): 1 usage
23     super().display()
24     status = "Required" if self.required_for_major else "Not Required"
25     print(f"- Core Course ({status} for Major)")
26
27 class ElectiveCourse(Course): 1 usage
28     def __init__(self, course_code, course_name, credit_hours, elective_type):
29         super().__init__(course_code, course_name, credit_hours)
30         self.elective_type = elective_type
```

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, search, and version control. The file explorer on the left lists several Python files: `in1.py`, `Tuple.py`, `functions.py`, `filehandling.py`, `Module.py`, `test_module.py`, `Oops.py`, `Encapsulation.py`, `Assignment6.py`, and `Assignment6(2).py`. The main editor displays the code for `Assignment6(2).py`, which defines two classes: `Course` and `CoreCourse`.

```
9 class Course: 2 usages
11     self.course_code = course_code
12     self.course_name = course_name
13     self.credit_hours = credit_hours
14
15     def display(self): 2 usages
16         print(f"{self.course_code}: {self.course_name} ({self.credit_hours} Credit Hours)")
17 class CoreCourse(Course): 1 usage
18     def __init__(self, course_code, course_name, credit_hours, required_for_major):
19         super().__init__(course_code, course_name, credit_hours)
20         self.required_for_major = required_for_major
21
22     def display(self): 1 usage
```

Below the editor, the Run console shows the output of the program. It displays the path to the Python interpreter and the output of the `display` method for three instances: `CS101`, `ENG201`, and an unnamed instance.

```
Run Assignment6(2) x
C:\Users\user\PycharmProjects\entri_d41_python_project\.venv\Scripts\python.exe "C:\Users\user\PycharmProjects\entri_d41_pyth
CS101: Introduction to Computer Science (3 Credit Hours)
- Core Course (Required for Major)
ENG201: Creative Writing (2 Credit Hours)
- Elective Course (Type: Liberal Arts)
```

The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for file operations, search, and running code. The file explorer on the left shows a project named 'entri_d41_python_project' with several files. The main editor displays a Python file named 'Assignment6.py' with the following code:

```
1 # Question 2: (5 Marks) Create a Python module named employee that contains a class Employee with attributes
2 # name, salary and methods get_name() and get_salary(). Write a program to use this module to create an object
3 # of the Employee class and display its name and salary.
4 class Employee: 2 usages
5     def __init__(self, name, salary):
6         self.name = name
7         self.salary = salary
8
9     def get_name(self): 1 usage
10        print(f"Employee Name: {self.name}")
11
12    def get_salary(self): 1 usage
13        print(f"Employee Salary: {self.salary}")
```

Below the editor, the 'Run' window shows the execution command and the output:

```
Run Assignment6 x
C:\Users\user\PycharmProjects\entri_d41_python_project\.venv\Scripts\python.exe C:\Users\user\PycharmProjects\entri_d41_pytho
Process finished with exit code 0
```

```
42  
43 from Assignment6 import Employee  
44 emp = Employee(name="Alice", salary=50000)  
45 emp.get_name()  
46 emp.get_salary()
```

Run Assignment6(2) x

Employee Name: Alice
Employee Salary: 50000

Process finished with exit code 0