

onlinegdb.com/online_python_compiler

Language Python 3

```
main.py
3 # Base class for Course
4 class Course:
5     def __init__(self, course_code, course_name, credit_hours):
6         self.course_code = course_code
7         self.course_name = course_name
8         self.credit_hours = credit_hours
9
10    def __str__(self):
11        return f"Course Code: {self.course_code}, Course Name: {self.course_name}, Credit Hours: {self.credit_hours}"
12
13 # SubClass for CoreCourse
14 class CoreCourse(Course):
15     def __init__(self, course_code, course_name, credit_hours, required_for_major):
16         super().__init__(course_code, course_name, credit_hours)
17         self.required_for_major = required_for_major
18
19     def __str__(self):
20        return super().__str__() + f", Required for Major: {self.required_for_major}"
21
22 # SubClass for ElectiveCourse
23 class ElectiveCourse(Course):
24     def __init__(self, course_code, course_name, credit_hours, elective_type):
25         super().__init__(course_code, course_name, credit_hours)
26         self.elective_type = elective_type
27
28     def __str__(self):
29        return super().__str__() + f", Elective Type: {self.elective_type}"
```

input

```
Course Code: CS101, Course Name: Introduction to Computer Science, Credit Hours: 3, Required for Major: True
Course Code: MATH101, Course Name: Calculus I, Credit Hours: 4, Required for Major: True
Course Code: HIST201, Course Name: World History, Credit Hours: 3, Elective Type: liberal arts
Course Code: PHYS101, Course Name: General Physics, Credit Hours: 3, Elective Type: technical
Course Code: ART101, Course Name: Introduction to Art, Credit Hours: 2, Elective Type: general
```

Employee Details:

```
Name: Alice, Salary: 50000
Name: Bob, Salary: 60000
Name: Charlie, Salary: 70000
Name: David, Salary: 55000
Name: Eva, Salary: 65000
```

1 33°C
Mostly sunny

Search Ps Chrome

12:39 23-03-2025

onlinegdb.com/online_python_compiler

Run | Debug | Stop | Share | Save | Beautify | ⚡

Language Python 3

main.py

```
31 # Creating instances for both Core and Elective courses
32 courses = [
33     CoreCourse("CS101", "Introduction to Computer Science", 3, True),
34     CoreCourse("MATH101", "Calculus I", 4, True),
35     ElectiveCourse("HIST201", "World History", 3, "liberal arts"),
36     ElectiveCourse("PHYS101", "General Physics", 3, "technical"),
37     ElectiveCourse("ART101", "Introduction to Art", 2, "general")
38 ]
39
40 # Displaying the course catalog
41 print("University Course Catalog:")
42 for course in courses:
43     print(course)
44
45 print("\n")
46
47 # Employee Management System - Question 2
48
49 # Employee Module (you could save this in a file called `employee.py`)
50 class Employee:
51     def __init__(self, name, salary):
52         self.name = name
53         self.salary = salary
54
55     def get_name(self):
56         return self.name
57
```

input

Course Code: CS101, Course Name: Introduction to Computer Science, Credit Hours: 3, Required for Major: True
Course Code: MATH101, Course Name: Calculus I, Credit Hours: 4, Required for Major: True
Course Code: HIST201, Course Name: World History, Credit Hours: 3, Elective Type: liberal arts
Course Code: PHYS101, Course Name: General Physics, Credit Hours: 3, Elective Type: technical
Course Code: ART101, Course Name: Introduction to Art, Credit Hours: 2, Elective Type: general

Employee Details:
Name: Alice, Salary: 50000
Name: Bob, Salary: 60000
Name: Charlie, Salary: 70000
Name: David, Salary: 55000
Name: Eva, Salary: 65000

1 33°C
Mostly sunny

Search



ENG
IN



12:39
23-03-2025

onlinegdb.com/online_python_compiler

Run | Debug | Stop | Share | Save | Beautify | Download

Language: Python 3

main.py

```
46
47 # Employee Management System - Question 2
48
49 # Employee Module (you could save this in a file called `employee.py`)
50 class Employee:
51     def __init__(self, name, salary):
52         self.name = name
53         self.salary = salary
54
55     def get_name(self):
56         return self.name
57
58     def get_salary(self):
59         return self.salary
60
61 # Creating Employee objects
62 employee_1 = Employee("Alice", 50000)
63 employee_2 = Employee("Bob", 60000)
64 employee_3 = Employee("Charlie", 70000)
65 employee_4 = Employee("David", 55000)
66 employee_5 = Employee("Eva", 65000)
67
68 # Displaying Employee details
69 employees = [employee_1, employee_2, employee_3, employee_4, employee_5]
70 print("Employee Details:")
71 for employee in employees:
72     print(f"Name: {employee.get_name()}, Salary: {employee.get_salary()}")
73
```

input

```
Course Code: CS101, Course Name: Introduction to Computer Science, Credit Hours: 3, Required for Major: True
Course Code: MATH101, Course Name: Calculus I, Credit Hours: 4, Required for Major: True
Course Code: HIST201, Course Name: World History, Credit Hours: 3, Elective Type: liberal arts
Course Code: PHYS101, Course Name: General Physics, Credit Hours: 3, Elective Type: technical
Course Code: ART101, Course Name: Introduction to Art, Credit Hours: 2, Elective Type: general

Employee Details:
Name: Alice, Salary: 50000
Name: Bob, Salary: 60000
Name: Charlie, Salary: 70000
Name: David, Salary: 55000
Name: Eva, Salary: 65000
```

33°C
Mostly sunny

Search

12:39
23-03-2025