MODULE 4

1. Write a program that has a class store which keeps a record of code and price of each product. Display a menu of all products to the user and prompt him to enter the quantity of each item required. Generate a bill and display the total amount.

```
class Store:
    def __init__(self):
        # code : (name, price)
        self.items = {
            101: ("Bread", 30),
            102: ("Milk", 25),
            103: ("Eggs", 6),
            104: ("Juice", 40)
        }

    def display(self):
        print("Code  Item    Price")
        for code, (name, price) in self.items.items():
            print(f"{code}   {name:<8} {price}")

    def bill(self):
        total = 0
        for code, (name, price) in self.items.items():
            qty = int(input(f"Enter quantity for {name}: "))
            total += qty * price
        print("\nTotal Bill Amount:", total)

s = Store()
s.display()
s.bill()
```

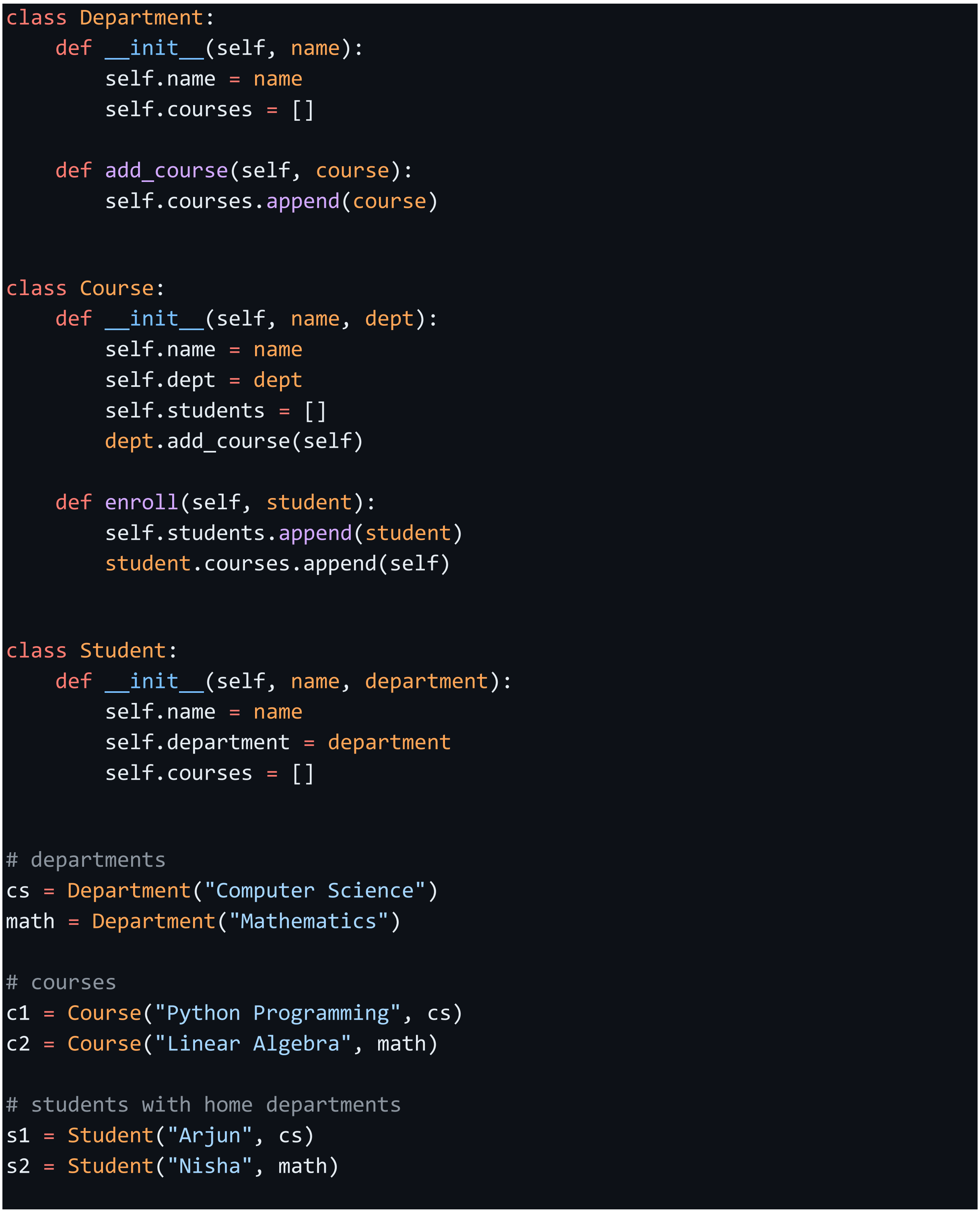
OUTPUT:

```
Code  Item    Price

101   Bread    30

102   Milk     25

103   Eggs     6

104   Juice    40

Enter quantity for Bread: 3

Enter quantity for Milk: 5

Enter quantity for Eggs: 20
```

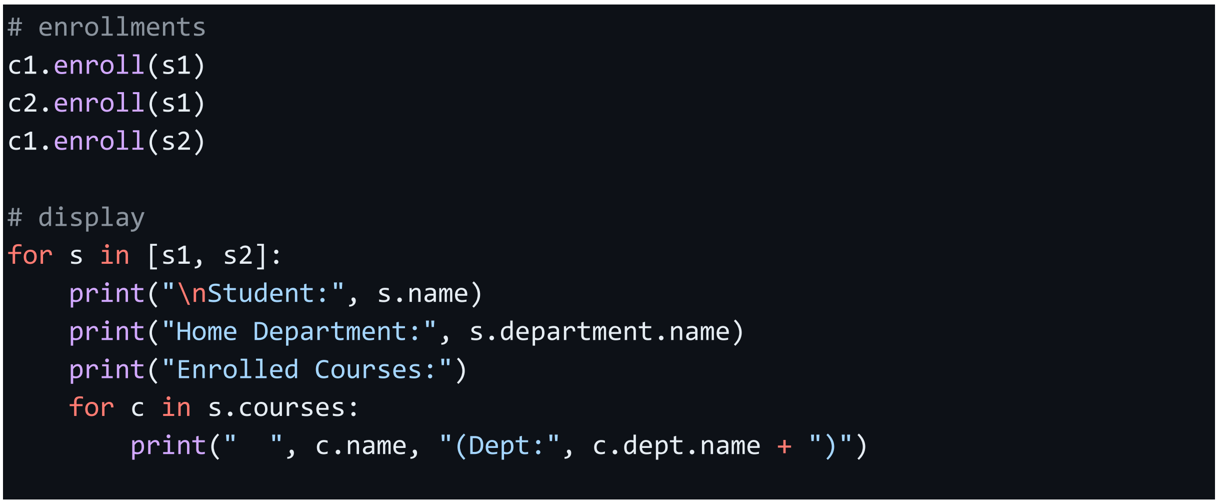
Enter quantity for Juice: 10

Total Bill Amount: 735

2.Write a program that has classes such as Student, Course and Department. Enroll a student in a course of a particular Department.



```
class Department:
    def __init__(self, name):
        self.name = name
        self.courses = []

    def add_course(self, course):
        self.courses.append(course)


class Course:
    def __init__(self, name, dept):
        self.name = name
        self.dept = dept
        self.students = []
        dept.add_course(self)

    def enroll(self, student):
        self.students.append(student)
        student.courses.append(self)


class Student:
    def __init__(self, name, department):
        self.name = name
        self.department = department
        self.courses = []


# departments
cs = Department("Computer Science")
math = Department("Mathematics")

# courses
c1 = Course("Python Programming", cs)
c2 = Course("Linear Algebra", math)

# students with home departments
s1 = Student("Arjun", cs)
s2 = Student("Nisha", math)
```

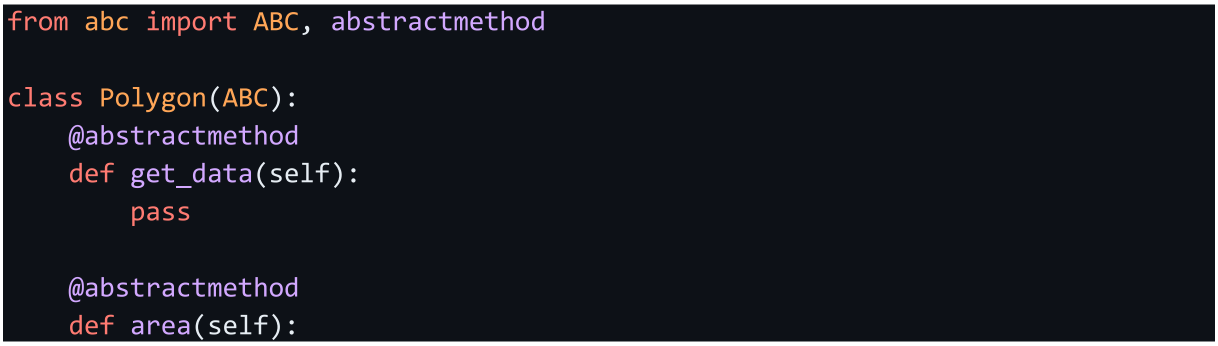


```python
# enrollments
c1.enroll(s1)
c2.enroll(s1)
c1.enroll(s2)

# display
for s in [s1, s2]:
    print("\nStudent:", s.name)
    print("Home Department:", s.department.name)
    print("Enrolled Courses:")
    for c in s.courses:
        print("  ", c.name, "(Dept:", c.dept.name + ")")
```

OUTPUT:

```
Student: Arjun
Home Department: Computer Science
Enrolled Courses:
   Python Programming (Dept: Computer Science)
   Linear Algebra (Dept: Mathematics)

Student: Nisha
Home Department: Mathematics
Enrolled Courses:
   Python Programming (Dept: Computer Science)
```

3.Write a program that has an abstract class polygon. Derive two classes Rectangle and Triangle from polygon and write methods to get the details of their dimensions and hence calculate the area.



```python
from abc import ABC, abstractmethod

class Polygon(ABC):
    @abstractmethod
    def get_data(self):
        pass

    @abstractmethod
    def area(self):
        pass


class Rectangle(Polygon):
    def get_data(self):
        self.l = float(input("Length: "))
        self.b = float(input("Breadth: "))

    def area(self):
        return self.l * self.b


class Triangle(Polygon):
    def get_data(self):
        self.b = float(input("Base: "))
        self.h = float(input("Height: "))

    def area(self):
        return 0.5 * self.b * self.h


# example
print("1. Rectangle\n2. Triangle")
ch = int(input("Enter choice: "))

if ch == 1:
    obj = Rectangle()
else:
    obj = Triangle()

obj.get_data()
print("Area =", obj.area())
```

OUTPUT:

```
1. Rectangle

2. Triangle

Enter choice: 2

Base: 30

Height: 50

Area = 750.0
```

-END-