Classes and Objects

1. Watch the video on creating a simple class. Create the class in the video and test it to make sure it works.

Add a method to the class that accepts a bonus rate for the employee. It should then compute the employee bonus of rate x salary and return this value. Demonstrate this method works by entering a bonus rate and displaying the bonus.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| Employee name | Class employee | Employee name |
| Employee salary | Def \_init\_(self, name, salary):  self.first = first  self.last = last  self.pay = pay  self.email = first + '.' + last + '@compay.com'  def fullname(self):  return '{} {}'.format(self.first, self.last)  def emailaddress(self):  return self.email  def bonus(self):  return self.pay \* 0.10 | Employee salary |
| Bonus rate | employee1 = Employee("Sam", "White", 50000)  employee2 = Employee("John", "Brown", 60000)  employee3 = Employee("Sarah", "Lee", 70000) | Bonus amount |
|  |  |  |
|  | for employee in [employee1, employee2, employee3]:  bonus = employee.bonus()  total\_salary = employee.pay + bonus  print("$", format(total\_salary, ".2f"), "is", employee.fullname(), f"({employee.email})'s salary with a 10% bonus.") |  |

1. Create a student class. This class should consist of student first name, student last name, district code (I or O) and enrolled credits. Create a method to compute tuition owed. Tuition owed should be enrolled credits times $250.00 per credit hour for in district students (district code of I) and or times $500.00 per credit hour for out of district students (district code of anything other than I). Test the class by instantiating the object and adding data.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| First name | Class student:  Def\_init\_(self, first name, last name, district code, enrolled credits)  Self,first name = first name  Self.last name = last name  Self.district code = district code  Self.enrolled credits = enrolled credits | Students’ information (first name, last name, district code, enrolled credits)  tuition owed |
| Last name | Def compute tuition (self):  If self.district code = I  Rate = 250  Else: rate = 500  Self.tuition = self.enrolled credits x rate  Return self.tuition |  |
| District code (I or O) |  |  |
| Enrolled credits | student\_1=student('Emma','Smith','I',15)  student\_2=student('Olivia','Brown','O',20)  student\_3=student('James','Miller','I',10) |  |
|  |  |  |
|  | print(f"{student\_1.first\_name} {student\_1.last\_name}, {student\_1.district\_code}, {student\_1.enrolled\_credits} credits, Tuition: ${student\_1.compute\_tuition()}")  print(f"{student\_2.first\_name} {student\_2.last\_name}, {student\_2.district\_code}, {student\_2.enrolled\_credits} credits, Tuition: ${student\_2.compute\_tuition()}")  print(f"{student\_3.first\_name} {student\_3.last\_name}, {student\_3.district\_code}, {student\_3.enrolled\_credits} credits, Tuition: ${student\_3.compute\_tuition()}") |  |