**Assignment Chapter 11 Randall Fishwick**

**CO SCI 124**

**Python Programming**

**Ex11\_2.py  
(Creates and uses an instance of the ShiftSupervisor object)**

# Program to use ShiftSupervisor Object  
  
import emp  
  
  
def main():  
 # Local variables  
 supervisor\_name = ''  
 supervisor\_id = ''  
 supervisor\_salary = 0.0  
 supervisor\_bonus = 0.0  
  
 # Get data attributes  
 supervisor\_name = input('\nEnter the name: ')  
 supervisor\_id = input('Enter the ID number: ')  
 supervisor\_salary = int(input('Enter the annual salary: $'))  
 supervisor\_bonus = float(input('Enter the annual bonus rate: $'))  
  
 # Create an instance of ShiftSupervisor  
 supervisor = emp.ShiftSupervisor(supervisor\_name, supervisor\_id, \  
 supervisor\_salary, supervisor\_bonus)  
  
 # Display information  
 print('\nShift supervisor information:')  
 print('Name:', supervisor.get\_name())  
 print('ID number:', supervisor.get\_id\_number())  
 print('Annual salary: $', format(supervisor.get\_annual\_salary(), ',.2f'), sep='')  
 print('Annual bonus: $', \  
 format(supervisor.get\_annual\_bonus(), ',.2f'), sep='')  
  
  
# Call the main function.  
main()

**Emp.py  
(Includes Employee Superclass, as well as ProductionWorker and ShiftSupervisor Subclasses)**

# Module emp  
  
# The Employee class represents a generic employee  
class Employee:  
 def \_\_init\_\_(self, name, id\_number):  
 self.\_\_name = name  
 self.\_\_id\_number = id\_number  
  
 def set\_name(self, name):  
 self.\_\_name = name  
  
 def set\_id\_number(self, id\_number):  
 self.\_\_id\_number = id\_number  
  
 def get\_name(self):  
 return self.\_\_name  
  
 def get\_id\_number(self):  
 return self.\_\_id\_number  
  
  
# The ProductionWorker class is a subclass of the Employee class  
class ProductionWorker(Employee):  
 def \_\_init\_\_(self, name, id\_number, shift\_number, pay\_rate):  
 # Call superclass \_\_init\_\_ method.  
 Employee.\_\_init\_\_(self, name, id\_number)  
  
 # Initialize the shift\_number and pay\_rate attributes.  
 self.\_\_shift\_number = shift\_number  
 self.\_\_pay\_rate = pay\_rate  
  
 # Mutator functions for shift\_number and pay\_rate.  
 def set\_shift\_number(self, shift\_number):  
 self.\_\_shift\_number = shift\_number  
  
 def set\_pay\_rate(self, pay\_rate):  
 self.\_\_pay\_rate = pay\_rate  
  
 # Accessor functions for shift\_number and pay\_rate.  
 def get\_shift\_number(self):  
 return self.\_\_shift\_number  
  
  
def get\_pay\_rate(self):  
 return self.\_\_pay\_rate  
  
  
# The ShiftSupervisor class is a subclass of the Employee class  
class ShiftSupervisor(Employee):  
 def \_\_init\_\_(self, name, id\_number, annual\_salary, annual\_bonus):  
 # Call superclass \_\_init\_\_ method.  
 Employee.\_\_init\_\_(self, name, id\_number)  
  
 # Initialize the annual\_salary and annual\_bonus attributes.  
 self.\_\_annual\_salary = annual\_salary  
 self.\_\_annual\_bonus = annual\_bonus  
  
 # Mutator functions for annual\_salary and annual\_bonus.  
 def set\_annual\_salary(self, annual\_salary):  
 self.\_\_annual\_salary = annual\_salary  
  
 def set\_annual\_bonus(self, annual\_bonus):  
 self.\_\_annual\_bonus = annual\_bonus  
  
 # Accessor functions for annual\_salary and annual\_bonus.  
 def get\_annual\_salary(self):  
 return self.\_\_annual\_salary  
  
 def get\_annual\_bonus(self):  
 return self.\_\_annual\_bonus