2] ShiftSupervisor Class

In a particular factory, a shift supervisor is a salaried employee who supervises a shift. In addition to a salary, the shift supervisor earns a yearly bonus when his or her shift meets production goals. Design a ShiftSupervisor class that is derived from the Employee class you created in the program made in the lab. The ShiftSupervisor class should have a member variable that holds the annual salary and a member variable that holds the annual production bonus that a shift supervisor has earned. Write one or more constructors and the appropriate accessor and mutator functions for the class.

Demonstrate the class by writing a program that uses a ShiftSupervisor object.

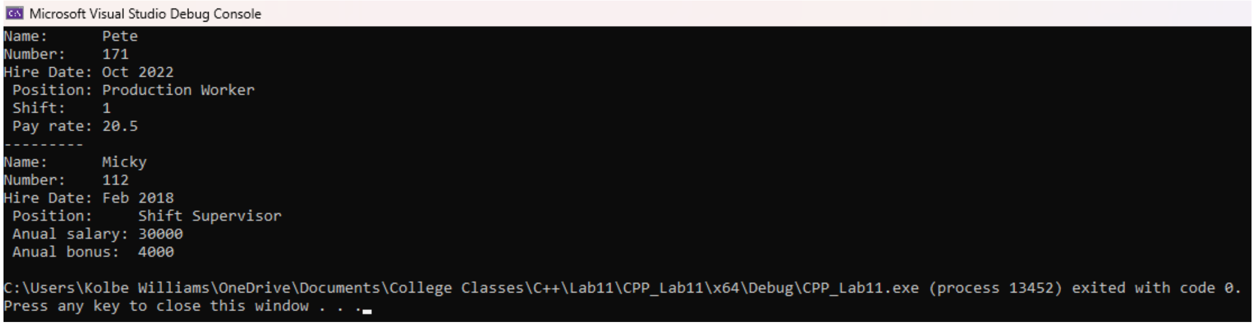
Source Code:

```
#include <iostream>
using namespace std;

class Employee
{
private:
	string employeeName, hireDate;
	int number;
public:
	Employee(string name, int num, string date) //Constructor
	{
		employeeName = name;
		number = num;
		hireDate = date;
	}

	void print() //Print function prints member fields
	{
		cout << "Name:      " << employeeName << endl;
		cout << "Number:    " << number << endl;
		cout << "Hire Date: " << hireDate << endl;
	}

	void setEmployeeName(string name) //Accessors and mutators for each field
	{ employeeName = name; }

	void setNumber(int num)
	{ number = num; }

	void setHireDate(string date)
	{ hireDate = date; }

	string getEmployeeName()
	{ return employeeName; }

	int getNumber()
	{ return number; }

	string getHireDate()
	{ return hireDate; }
};
```

```
class ProductionWorker : public Employee //Class is derived from Employee class
{
private:
	int shift;
	double hourlyPayRate;
public:
	//Constructor uses base class constructor
	ProductionWorker(string name, int num, string date, int shiftNum, double pay)
: Employee(name, num, date)
	{
		shift = shiftNum;
		hourlyPayRate = pay;
	}

	void print() //Print function prints member fields
	{
		Employee::print();
		cout << " Position: Production Worker\n";
		cout << " Shift:    " << shift << endl;
		cout << " Pay rate: " << hourlyPayRate << endl;
	}

	void setShift(int shiftNum) //Accessors and mutators for each field
	{ shift = shiftNum; }

	void setHourlyPayRate(double pay)
	{ hourlyPayRate = pay; }

	int getShift()
	{ return shift; }

	double getHourlyPayRate()
	{ return hourlyPayRate; }
};

class ShiftSupervisor : public Employee //Class is derived from Employee class
{
private:
	int anualSalary, anualProductionBonus;
public:
	//Constructor uses base class constructor
	ShiftSupervisor(string name, int num, string date, int salary, int bonus) :
Employee(name, num, date)
	{
		anualSalary = salary;
		anualProductionBonus = bonus;
	}

	void print() //Print function prints member fields
	{
		Employee::print();
		cout << " Position:     Shift Supervisor\n";
		cout << " Anual salary: " << anualSalary << endl;
		cout << " Anual bonus:  " << anualProductionBonus << endl;
	}

	void setAnualSalary(int salary) //Accessors and mutators for each field
```

```
    { anualSalary = salary; }

    void setAnualProductionBonus(int bonus)
    { anualProductionBonus = bonus; }

    int getAnaulSalary()
    { return anualSalary; }

    int getAnaulProductionBonus()
    { return anualProductionBonus; }
};

int main()
{
    ProductionWorker pw("Pete", 171, "Oct 2022", 1, 20.5);
    pw.print();
    cout << "---------" << endl;
    ShiftSupervisor ss("Micky", 112, "Feb 2018", 30000, 4000);
    ss.print();
    return 0;
}
```

Output:



```
Name:      Pete
Number:    171
Hire Date: Oct 2022
 Position: Production Worker
 Shift:    1
 Pay rate: 20.5
---------
Name:      Micky
Number:    112
Hire Date: Feb 2018
 Position:     Shift Supervisor
 Anual salary: 30000
 Anual bonus:  4000

C:\Users\Kolbe Williams\OneDrive\Documents\College Classes\C++\Lab11\CPP_Lab11\x64\Debug\CPP_Lab11.exe (process 13452) exited with code 0.
Press any key to close this window . . .
```