TX00FL42-3001

INHERITANCE

MUATH OTHMAN

LET’S CHECK HOMEWORKS!

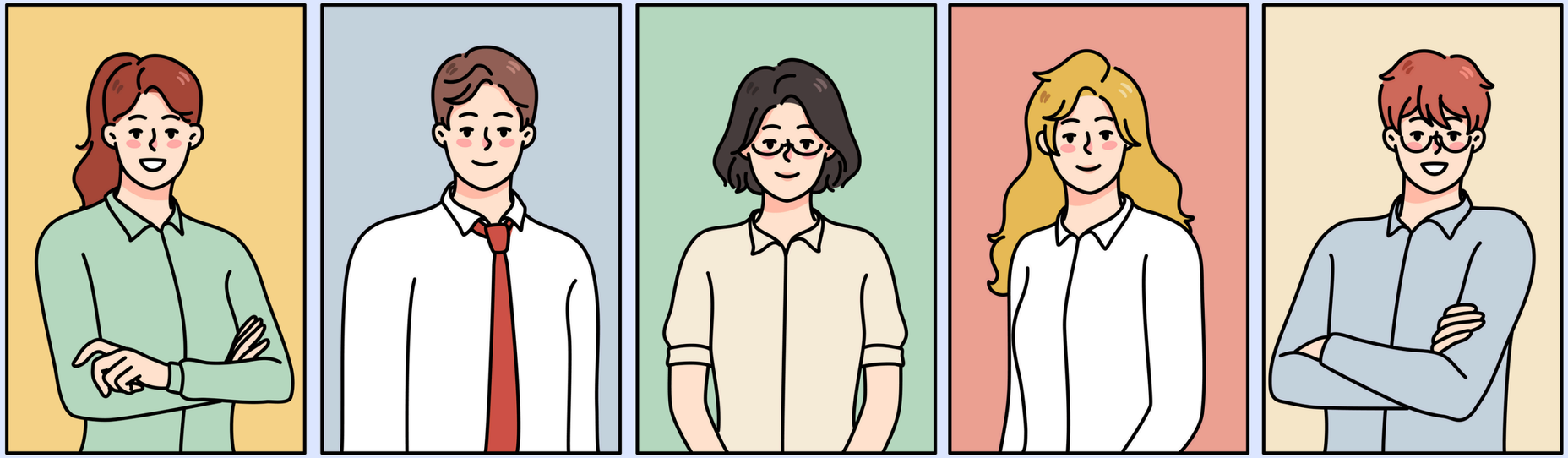
LET’S DO TOGETHER EXERCISE 4!

WHAT IS INHERITANCE?

Inheritance is a fundamental concept in object-oriented programming where one class, known as the base class (or parent class), provides attributes and methods that are shared with another class, known as the subclass (or child class). This structure creates a hierarchical relationship between classes, allowing subclasses to use and extend the functionality of the base class.

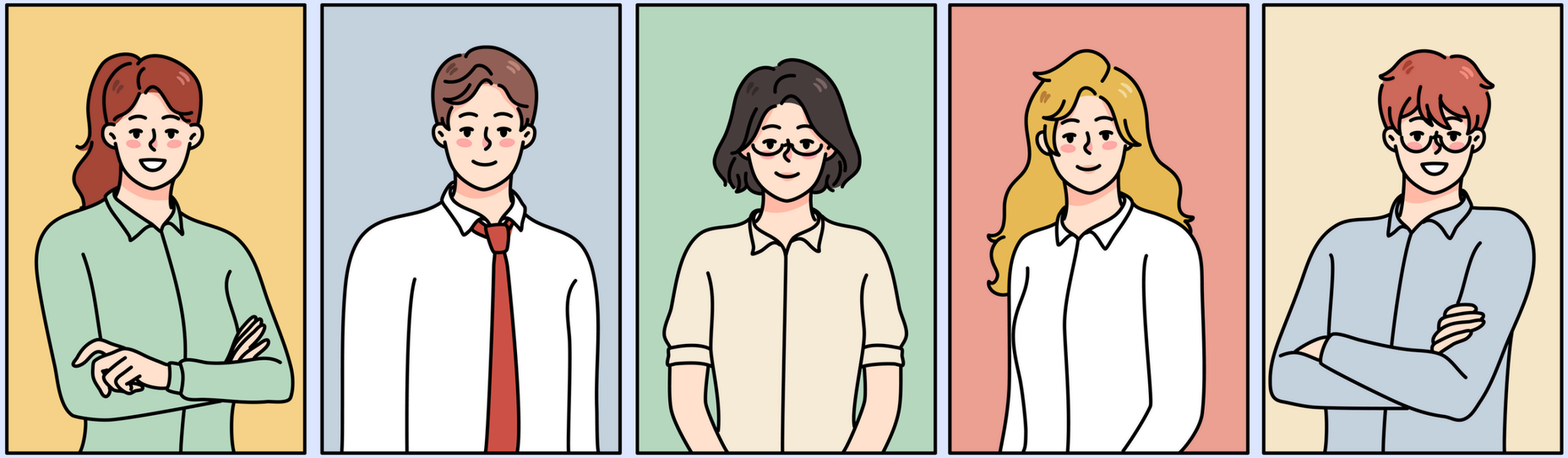
LET’S TAKE EXAMPLE

Hourly Hourly Month Hourly Month



EMPLOYEES

Hourly Hourly Month Hourly Month



EXAMPLE: BASE CLASS

```python
class Employee:

    total_employees = 0

    def __init__(self, first_name, last_name):
        Employee.total_employees = Employee.total_employees + 1
        self.employee_number = Employee.total_employees
        self.first_name = first_name
        self.last_name = last_name

    def print_information(self):
        print(f"{self.employee_number}: {self.first_name} {self.last_name}")
```

EXAMPLE: MAIN

```
employees = []
employees.append(Employee("Viivi", "Virta"))
employees.append(Employee("Ahmed", "Habib"))

for e in employees:
    e.print_information()
```

EXAMPLE: SUBCLASSES

```python
class HourlyPaid(Employee):

    def __init__(self, first_name, last_name, hourly_pay):
        self.hourly_pay = hourly_pay
        super().__init__(first_name, last_name)




class MonthlyPaid(Employee):

    def __init__(self, first_name, last_name, monthly_pay):
        self.monthly_pay = monthly_pay
        super().__init__(first_name, last_name)
```

EXAMPLE: MAIN

```
employees = []
employees.append(HourlyPaid("Viivi", "Virta", 12.35))
employees.append(MonthlyPaid("Ahmed", "Habib", 2750))
employees.append(Employee("Pekka", "Puro"))
employees.append(HourlyPaid("Olga", "Glebova", 14.92))

for e in employees:
    e.print_information()
```

EXERCISE

EXAMPLE: METHOD OVERRIDING

```
def print_information(self):
    super().print_information()
    print(f"Hourly pay: {self.hourly_pay}")
```

EXAMPLE: SUBCLASSES

```python
class HourlyPaid(Employee):

    def __init__(self, first_name, last_name, hourly_pay):
        self.hourly_pay = hourly_pay
        super().__init__(first_name, last_name)

    def print_information(self):
        super().print_information()
        print(f"Hourly pay: {self.hourly_pay}")

class MonthlyPaid(Employee):

    def __init__(self, first_name, last_name, monthly_pay):
        self.monthly_pay = monthly_pay
        super().__init__(first_name, last_name)

    def print_information(self):
        super().print_information()
        print(f"Monthly pay: {self.monthly_pay}")
```

EXERCISE

EXAMPLE: MULTIPLE INHERITANCE

```python
class Vehicle:
    def __init__(self, speed):
        self.speed = speed

class SportsItem:
    def __init__(self, weight):
        self.weight = weight

class Bicycle(Vehicle, SportsItem):
    def __init__(self, speed, weight, gears):
        Vehicle.__init__(self, speed)
        SportsItem.__init__(self, weight)
        self.gears = gears
```

EXAMPLE: MAIN

```
b = Bicycle(45, 18.7, 3)
print(b.gears)
print(b.speed)
print(b.weight)
```

EXAMPLE: INITIALIZER

```
# Incorrect initializer calls
super.__init__(speed)
super.__init__(weight)


# Correct initializer calls
Vehicle.__init__(self, speed)
SportsItem.__init__(self, weight)
```

EXERCISE