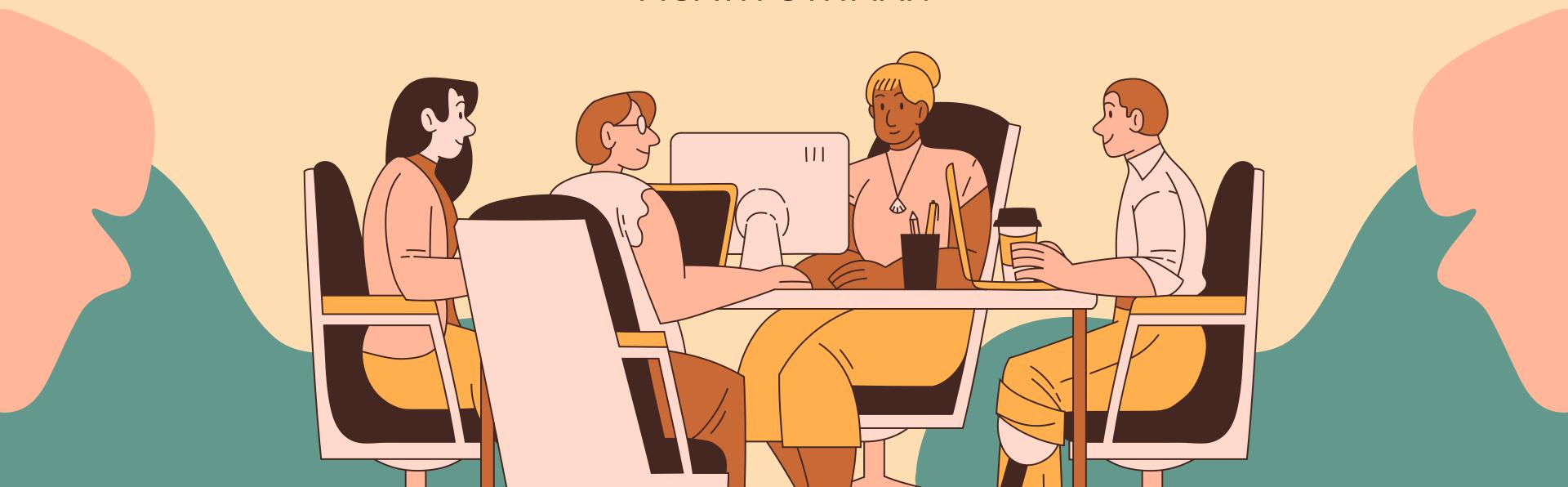
TX00FL42-3001

INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING

MUATH OTHMAN



LET'S CHECK HOMEWORKS!



WHATIS CLASS?



A class is a general concept defining common properties and actions for its objects.

LET'S TAKE EXAMPLE





INTERNATIONAL PASSPORT



Surname
MUATH
Given Names
OTHMAN
Date of birth
20-11-1995
Date of issue

01-09-2024 Date of expiry 01-09-2028 Passport No. AA018465

Personal No.

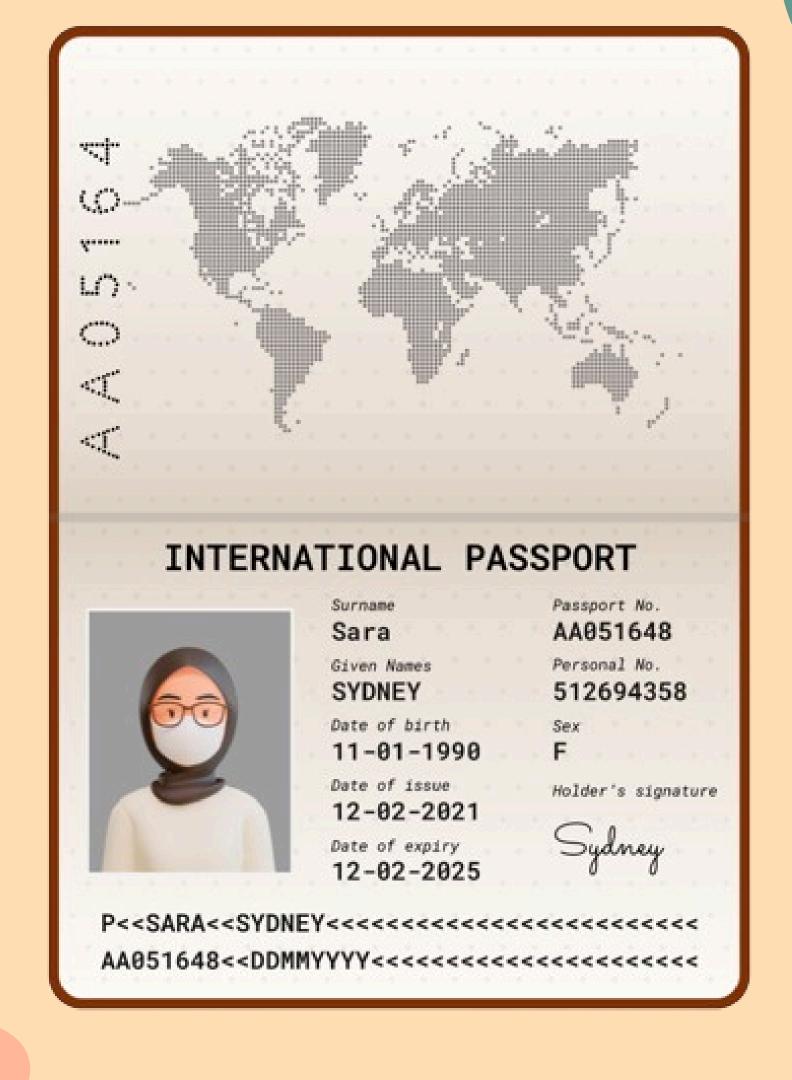
012684529

Sex

M

Holder's signature

muath Othman



WHAT PROPERTIES DOES A PASSPORT HAVE?

WHAT ACTIONS DOES A PASSPORT HAVE?





The pass statement is a placeholder to create an empty class without properties or methods.

```
class Passport:
    pass

passport = Passport()
passport.passport_number = "A1234567"
passport.nationality = "Canadian"

print(f"Passport {passport_number} for a citizen of {passport.nationality}.")
```

NAMING CONVENTIONS FOR CLASSES

- Class names should start with an uppercase letter.
- Multiple words in a class name should be joined together in CamelCase.
- Example: ScreenRectangle.

EXERCISE

1. Create a class named Car.

 Use the pass statement to define it without any attributes or methods.

```
class Passport:
    def __init__(self, passport_number, nationality):
        self.passport_number = passport_number
        self.nationality = nationality

passport = Passport("A12345678", "Canadian")
print(f"Passport {passport.passport_number} for a citizen of {passport.nationality}.")
```

INITIALIZER PROPER USAGE

```
#Incorrect
passport = Passport(self, "A12345678", "Canadian")

#Correct
passport = Passport("A12345678", "Canadian")
```

EXERCISE

2. Adding a Constructor with Initial Attributes.

• Extend the Car class with a constructor method to initialize attributes.

2. Adding a Constructor with Initial Attributes.

```
# Example code:
car1 = Car("Toyota", "Corolla", 2020, "Red")
print(car1.make) # Should print "Toyota"
print(car1.model) # Should print "Corolla"
print(car1.year) # Should print 2020
print(car1.color) # Should print "Red"
```

DEFINING METHODS FOR ACTIONS

```
class Passport:
    def __init__(self, passport_number, nationality, location_status = True):
        self.passport_number = passport_number
        self.nationality = nationality
        self.location_status = location_status
    def update_travel_status(self, is_inside):
        self.location_status = is_inside
# Create a Passport instance
passport1 = Passport("A1234567", "Canadian")
# Print initial status
print(f"Is the passport holder inside the country? {passport1.location_status}")
# Setting status to "outside the country"
passport1.update_travel_status(False)
# Print new travel status
print(f"Is the passport holder inside the country? {passport1.location_status}")
```

DEFAULT PARAMETER VALUES

```
# Create two Passport instances

passport1 = Passport("A1234567", "Canadian") # Uses default location_status (inside the country)

passport2 = Passport("B7654321", "American", False) # Custom location_status (outside the country)
```

CLASS VARIABLES OR STATIC VARIABLES

```
class Passport:
    created = 0 # Class variable to count the number of Passport instances
    def __init__(self, passport_number, nationality, location_status=True):
        self.passport_number = passport_number
        self.nationality = nationality
        self.location_status = location_status
        Passport.created += 1 # Increment the count whenever a new Passport is created
# Create Passport instances
passport1 = Passport("A1234567", "Canadian")
passport2 = Passport("B7654321", "American", False)
# Print the number of passports created
print(f"{Passport.created} passports have been created so far.")
```

STUDENTS WHO DID NOT PASS THE EXAM SHOULD STAY WITH ME AFTER THE LECTURE.

EXERCISE

3. Adding Methods and Class Variables.

Define a class variable total_cars

 Add a method car_age that calculates the age of the car based on the current year