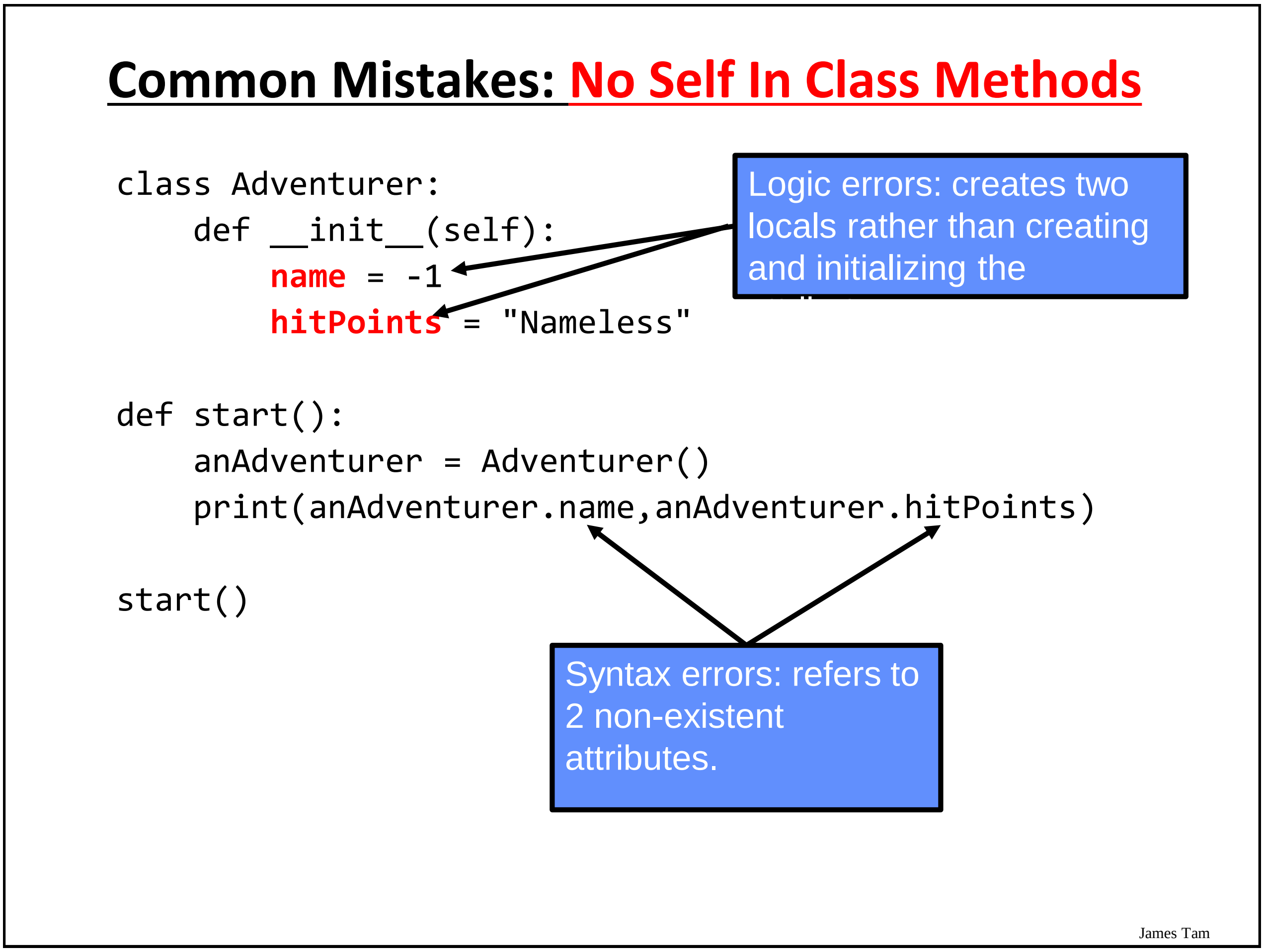
Exercise 1

- Define a class ‘Adventurer’ that consists of two attributes:
  - Hit points: a whole number value.
  - Name
- As instances of the class are created these attributes will be set to these initial values:
  - Hit points are negative one.
  - The name is “Nameless”
- Define and call a starting function.
  - An instance of an adventurer will be created.
  - The attributes will be displayed onscreen.

James Tam

Solution

```
class Adventurer:
    def __init__(self):
        self.name = -1
        self.hitPoints = "Nameless"

def start():
    anAdventurer = Adventurer()
    print(anAdventurer.name,anAdventurer.hitPoints)

start()
```



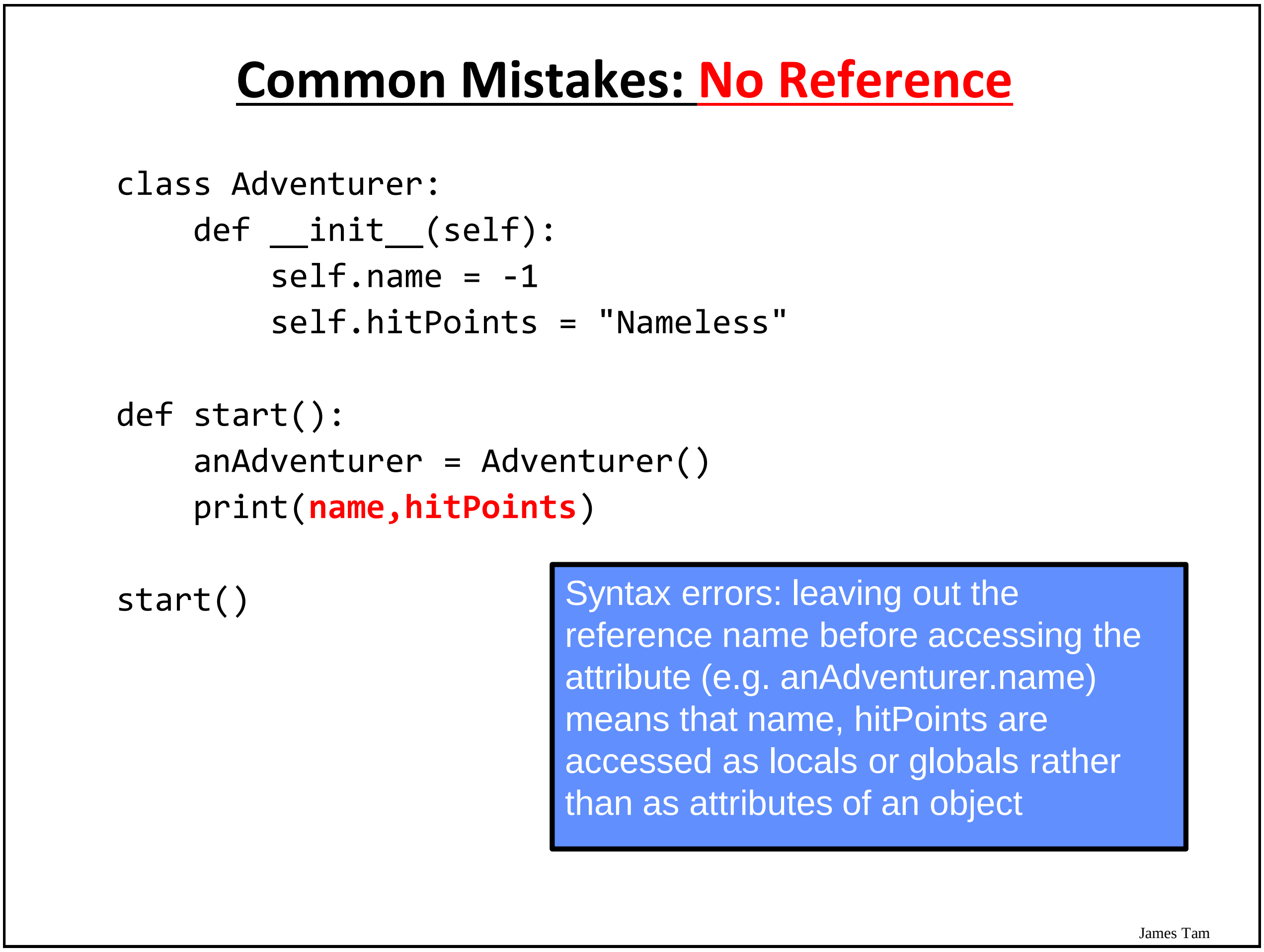
Common Mistakes: No Self In Class Methods

```
class Adventurer:
    def __init__(self):
        name = -1
        hitPoints = "Nameless"

def start():
    anAdventurer = Adventurer()
    print(anAdventurer.name,anAdventurer.hitPoints)

start()
```

Logic errors: creates two locals rather than creating and initializing the

Syntax errors: refers to 2 non-existent attributes.

James Tam

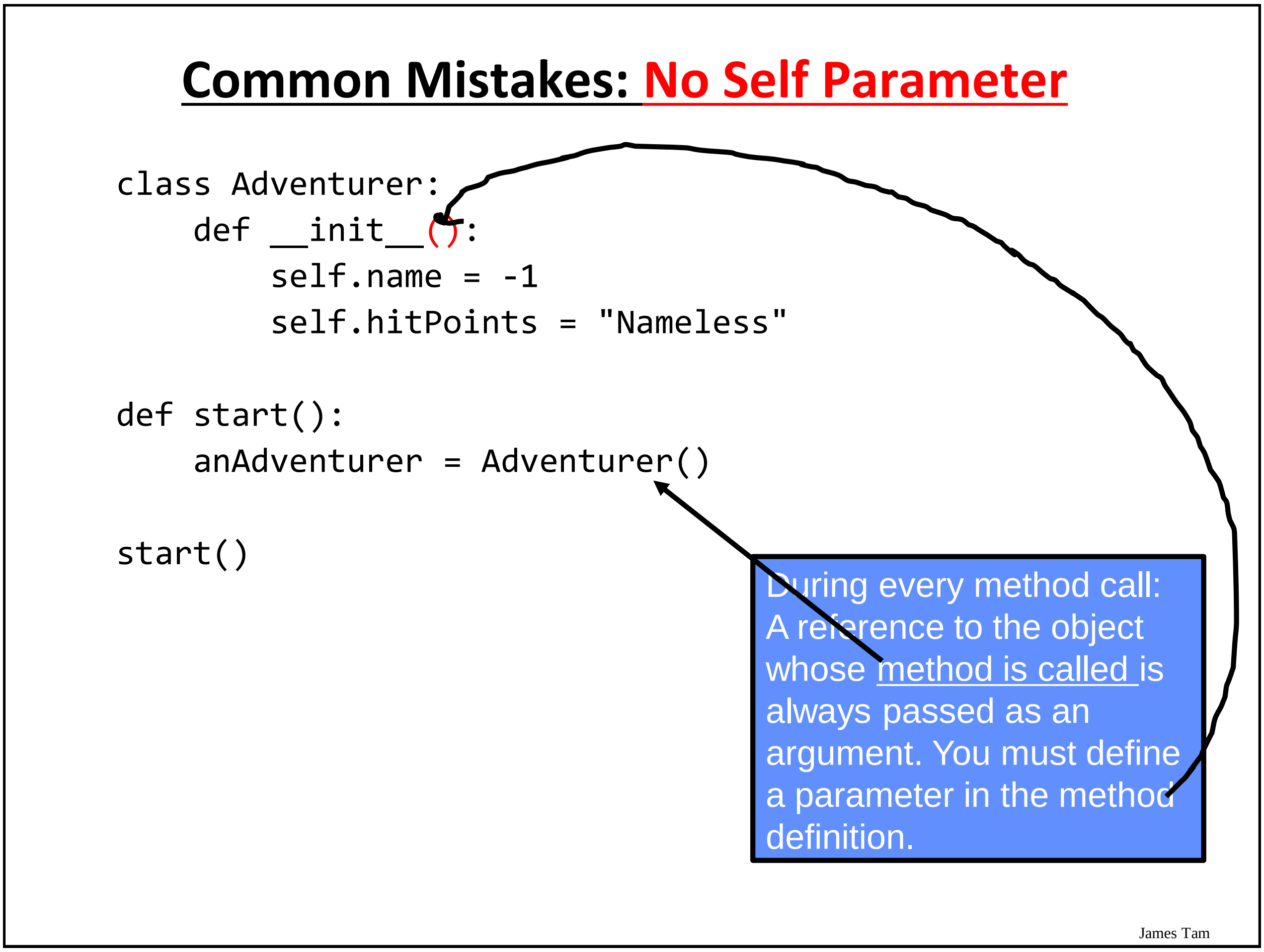


Common Mistakes: No Reference

```
class Adventurer:
    def __init__(self):
        self.name = -1
        self.hitPoints = "Nameless"

def start():
    anAdventurer = Adventurer()
    print(name,hitPoints)

start()
```

Syntax errors: leaving out the reference name before accessing the attribute (e.g. anAdventurer.name) means that name, hitPoints are accessed as locals or globals rather than as attributes of an object

James Tam



Common Mistakes: No Self Parameter

```
class Adventurer:
    def __init__():
        self.name = -1
        self.hitPoints = "Nameless"

def start():
    anAdventurer = Adventurer()

start()
```

During every method call: A reference to the object whose method is called is always passed as an argument. You must define a parameter in the method definition.

James Tam