

```
python assignment 3 > 🐍 python assignment 3.py > ...
1 # Creating a parent class named 'animal' with a method 'speak'
2 ✓ class animal:
3   ✓ def speak(self):
4     |   print("parent element animal speaking")
5 # Creating a child class named 'tiger' that inherits from 'animal' and overrides the 'speak'
6   method
6 ✓ class tiger(animal):
7   ✓ def speak(self):
8     |   print("child element tiger speaking")
9 # Creating instances of both classes and calling their 'speak' methods
10 t = tiger()
11 a = animal()
12 # Calling the speak method on both instances
13 t.speak()
14 a.speak()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Lenovo\OneDrive\Documents\GitHub> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python312
/python.exe "c:/Users/Lenovo/OneDrive/Documents/GitHub/python assignment 3/python assignment 3.py"
child element tiger speaking
parent element animal speaking
PS C:\Users\Lenovo\OneDrive\Documents\GitHub>
```

```
python assignment 3 > Method Overloading.py > ...
1 # class with method overloading using default parameters
2 class animal:
3     # __init__ method with default parameters to simulate method overloading
4     def __init__(self, name="unknown", species="unknown"):
5         self.name = name
6         self.species = species
7         print(f"Animal created: Name = {self.name}, Species = {self.species}")
8 # Creating instances of the animal class with different parameters
9 a = animal()
10 b = animal('tiger')
11 c = animal('lion', 'Mammelian')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python + ⌂ ⌄ ... | ⌂ X

```
PS C:\Users\Lenovo\OneDrive\Documents\GitHub> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/lenovo/OneDrive/Documents/GitHub/python assignment 3/Method Overloading.py"
Animal created: Name = unknown, Species = unknown
Animal created: Name = tiger, Species = unknown
Animal created: Name = lion, Species = Mammelian
PS C:\Users\Lenovo\OneDrive\Documents\GitHub>
```

```
python assignment 3 > types of constructors.py > ...
1 # creating a class student with parameterized constructor
2 v class student :
3     #parameterized constructor
4 v     def __init__(self,name,age):
5         self.name = name
6         self.age = age
7         print(f" Student name : {self.name} Student age : {self.age}")
8
9 v     def display(self):
10        print(f" Student name : {self.name} Student age : {self.age}")
11 # creating an object of the class student which invokes the parameterized constructor
12 s1 = student("abc",21)
13 # calling the display method to show student details
14 s1.display()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python + ▾ ⊞ ✎ ⋮ | [] ×

```
PS C:\Users\Lenovo\OneDrive\Documents\GitHub> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Lenovo/OneDrive/Documents/GitHub/python assignment 3/types of constructors.py"
Student name : abc Student age : 21
Student name : abc Student age : 21
PS C:\Users\Lenovo\OneDrive\Documents\GitHub>
```