

Not syncing

Home

assignment file handling, exce

GitHub - Search

SEENIYA24/Python-Modules: F

localhost:8888/notebooks/python%2Fassignment%20file%20handling%2C%20exception%20handling%2Coopsitled9.ipynb

Jupyter

assignment file handling, exception handling,oopsitled9

Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Trusted

JupyterLab Python 3 (ipykernel)

```
[ ]: #assignment file handling,exception handling,oops

[ ]: #1.Write a Python program to read a file and display its contents

[1]: def read_and_display_file(filename):
      try:
          with open(filename, 'r') as file:
              contents = file.read()
              print(contents)
      except FileNotFoundError:
          print(f"Error: File '{filename}' not found.")
      except IOError:
          print(f"Error: Unable to read file '{filename}'.")

      # Example usage
      filename = "example.txt"
      read_and_display_file(filename)

      Error: File 'example.txt' not found.

[ ]:

[ ]: #2.Write a Python program to copy the contents of one file to another file

[3]: def copy_file(source_file, destination_file):
      try:
          with open(source_file, 'r') as source:
              contents = source.read()

          with open(destination_file, 'w') as destination:
              destination.write(contents)
```

27°

ENG IN 10:09 PM 20-10-2024

Not syncing Home Home assignment file handling, exce GitHub - Search SEENIYA24/Python-Modules: F

localhost:8888/notebooks/python%2Fassignment%20file%20handling%2C%20exception%20handling%2Coopsitled9.ipynb

Jupyter assignment file handling, exception handling,oopsitled9 Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

```
[ ]:
[ ]: #2. Write a Python program to copy the contents of one file to another file
[3]: def copy_file(source_file, destination_file):
      try:
          with open(source_file, 'r') as source:
              contents = source.read()

          with open(destination_file, 'w') as destination:
              destination.write(contents)

          print(f"Contents of '{source_file}' successfully copied to '{destination_file}'.")
      except FileNotFoundError:
          print(f"Error: Source file '{source_file}' not found.")
      except IOError:
          print("Error: Unable to read source file or write to destination file.")

      # Example usage
      source = "source.txt"
      destination = "destination.txt"
      copy_file(source, destination)

      Error: Source file 'source.txt' not found.
[ ]: #3. Write a Python program to read the content of a file and count the total number of words in that file.
[5]: def count_words_in_file(filename):
      try:
          with open(filename, 'r') as file:
              content = file.read()
              words = content.split()
```

27° 10:09 PM 20-10-2024

Not syncing

Home

Home


assignment file handling, exce

GitHub - Search

SEENIYA24/Python-Modules: F

+

localhost:8888/notebooks/python%2Fassignment%20file%20handling%2C%20exception%20handling%2Coopsitled9.ipynb

 **jupyter** assignment file handling, exception handling,oopsitled9 Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Trusted

JupyterLab Python 3 (ipykernel)

```
destination = "destination.txt"
copy_file(source, destination)

Error: Source file 'source.txt' not found.

[ ]: #3. Write a Python program to read the content of a file and count the total number of words in that file.

[5]: def count_words_in_file(filename):
    try:
        with open(filename, 'r') as file:
            content = file.read()
            words = content.split()
            word_count = len(words)
            print(f"The file '{filename}' contains {word_count} words.")
        return word_count
    except FileNotFoundError:
        print(f"Error: File '{filename}' not found.")
    except IOError:
        print(f"Error: Unable to read file '{filename}'.")


    # Example usage
    filename = "example.txt"
    count_words_in_file(filename)

Error: File 'example.txt' not found.

[ ]: #4. Write a Python program to read the content of a file and count the number of occurrences of a specific word in that file

[7]: def count_word_occurrences(filename, target_word):
    try:
        with open(filename, 'r') as file:
            content = file.read()
            words = content.lower().split()
            word_count = words.count(target_word.lower())
            print(f"The word '{target_word}' appears {word_count} times in the file '{filename}'.")
```

27°



ENG IN 10:10 PM 20-10-2024

```
filename = "example.txt"
count_words_in_file(filename)
```

```
Error: File 'example.txt' not found.
```

```
[ ]: #4. Write a Python program to read the content of a file and count the number of occurrences of a specific word in that file
```

```
[7]: def count_word_occurrences(filename, target_word):
    try:
        with open(filename, 'r') as file:
            content = file.read()
            words = content.lower().split()
            word_count = words.count(target_word.lower())
            print(f"The word '{target_word}' appears {word_count} times in the file '{filename}'.")
            return word_count
    except FileNotFoundError:
        print(f"Error: File '{filename}' not found.")
    except IOError:
        print(f"Error: Unable to read file '{filename}'.")
```

Example usage

```
filename = "example.txt"
target_word = "python"
count_word_occurrences(filename, target_word)
```

```
Error: File 'example.txt' not found.
```

[]: #5. Write a Python program that prompts the user to input a string and converts it to an integer. Use try-except blocks to handle any exceptions that might occur.

```
[9]: def string_to_int():
    try:
        user_input = input("Enter a number: ")
        number = int(user_input)
        print(f"Successfully converted '{user_input}' to integer: {number}")
```


Not syncing

Home

assignment file handling, exce

GitHub - Search

SEENIYA24/Python-Modules: F

localhost:8888/notebooks/python%2Fassignment%20file%20handling%20C%20exception%20handling%20oopsitled9.ipynb

Jupyter assignment file handling, exception handling,oopsitled9 Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Trusted

JupyterLab Python 3 (ipykernel)

```
target_word = python
count_word_occurrences(filename, target_word)

Error: File 'example.txt' not found.

[ ]: #5. Write a Python program that prompts the user to input a string and converts it to an integer. Use try-except blocks to handle any exceptions that might occur.

[9]: def string_to_int():
    try:
        user_input = input("Enter a number: ")
        number = int(user_input)
        print(f"Successfully converted '{user_input}' to integer: {number}")
        return number
    except ValueError:
        print(f"Error: '{user_input}' cannot be converted to an integer.")

# Example usage
string_to_int()

Enter a number: 10
Successfully converted '10' to integer: 10

[9]: 10

[ ]: #6. Write a Python program that prompts the user to input a list of integers and raises an exception if any of the integers in the list are negative.

[11]: class NegativeIntegerError(Exception):
    pass

    def check_negative_integers():
        try:
            input_string = input("Enter a list of integers separated by spaces: ")
            integer_list = [int(num) for num in input_string.split()]
            # Check for negative integers
            for num in integer_list:
                if num < 0:
                    raise NegativeIntegerError(f"Negative integer found: {num}")
            print("All integers are non-negative.")
        except NegativeIntegerError as e:
            print(e)

# Example usage
check_negative_integers()
```

27°

ENG IN 10:10 PM 20-10-2024

Not syncing

Home

assignment file handling, exce

GitHub - Search

SEENIYA24/Python-Modules: F

localhost:8888/notebooks/python%2Fassignment%20file%20handling%2C%20exception%20handling%2Coopsitled9.ipynb

Jupyter assignment file handling, exception handling,oopsitled9 Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Trusted

JupyterLab Python 3 (ipykernel)

```
[ ]: #6. Write a Python program that prompts the user to input a list of integers and raises an exception if any of the integers in the list are negative.

[11]: class NegativeIntegerError(Exception):
      pass

      def check_negative_integers():
          try:
              input_string = input("Enter a list of integers separated by spaces: ")
              integer_list = [int(num) for num in input_string.split()]

              for num in integer_list:
                  if num < 0:
                      raise NegativeIntegerError(f"Negative integer found: {num}")

              print("All integers are non-negative.")
              return integer_list
          except ValueError:
              print("Error: Invalid input. Please enter integers separated by spaces.")
          except NegativeIntegerError as e:
              print(f"Error: {e}")

      # Example usage
      check_negative_integers()

      Enter a list of integers separated by spaces: -4
      Error: Negative integer found: -4

[ ]: #7. Write a Python program that prompts the user to input a list of integers and computes the average of those integers. Use try-except blocks to handle a

[13]: def compute_average():
      try:
          input_string = input("Enter a list of integers separated by spaces: ")
```

27°

ENG IN 10:10 PM 20-10-2024

Not syncing

Home

assignment file handling, exce

GitHub - Search

SEENIYA24/Python-Modules: F

localhost:8888/notebooks/python%2Fassignment%20file%20handling%2C%20exception%20handling%2Coopsitled9.ipynb

Jupyter

assignment file handling, exception handling,oopsitled9

Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Trusted

JupyterLab Python 3 (ipykernel)

Error: Negative integer found: -4

[]:

#7. Write a Python program that prompts the user to input a list of integers and computes the average of those integers. Use try-except blocks to handle a

[13]:

```
def compute_average():
    try:
        input_string = input("Enter a list of integers separated by spaces: ")
        integer_list = [int(num) for num in input_string.split()]

        if not integer_list:
            raise ValueError("The list is empty.")

        average = sum(integer_list) / len(integer_list)
        print(f"The average of the integers is: {average}")
        return average
    except ValueError as e:
        print(f"Error: {e}")
    except ZeroDivisionError:
        print("Error: Cannot compute average of an empty list.")
    finally:
        print("Program has finished running.")

# Example usage
compute_average()
```

Enter a list of integers separated by spaces: 9,-4

Error: invalid literal for int() with base 10: '9,-4'

Program has finished running.

[]:

#8. Write a Python program that prompts the user to input a filename and writes a string to that file. Use try-except blocks to handle any exceptions that

[17]:

```
def write_to_file():
    try:
        filename = input("Enter the filename: ")
        content = input("Enter the content to write to the file: ")
```

27°

Windows Taskbar

10:10 PM 20-10-2024

Not syncing

Home

Home

assignment file handling, exce

GitHub - Search

SEENIYA24/Python-Modules: F

localhost:8888/notebooks/python%2Fassignment%20file%20handling%20C%20exception%20handling%20oopsitled9.ipynb

Jupyter assignment file handling, exception handling,oopsitled9 Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Trusted

JupyterLab Python 3 (ipykernel)

[]: #8. Write a Python program that prompts the user to input a filename and writes a string to that file. Use try-except blocks to handle any exceptions that

[17]:

```
def write_to_file():
    try:
        filename = input("Enter the filename: ")
        content = input("Enter the content to write to the file: ")

        with open(filename, 'w') as file:
            file.write(content)

        print(f"Successfully wrote content to '{filename}'.")
        print("Welcome! File operation completed successfully.")
    except IOError:
        print(f"Error: Unable to write to file '{filename}'.")

# Example usage
write_to_file()
```

Enter the filename: seeniya
Enter the content to write to the file: assignment python
Successfully wrote content to 'seeniya'.
Welcome! File operation completed successfully.

[]: #9. Build a program to manage a university's course catalog. You want to define a base class Course that has the following properties:

[19]:

```
class Course:
```

course_code: a string representing the course code (e.g., "CS101")
course_name: a string representing the course name (e.g., "Introduction to Computer Science")
credit_hours: an integer representing the credit hours for the course (e.g., 3)
You also want to define two subclasses CoreCourse and ElectiveCourse, which inherit from the Course class.
CoreCourse should have an additional property required_for_major which is a boolean representing whether the course is required for a particular major.
ElectiveCourse should have an additional property elective_type which is a string representing the type of elective (e.g., "general", "technical", "liber.

Not syncing

Home

assignment file handling, exce

GitHub - Search

SEENIYA24/Python-Modules: F

localhost:8888/notebooks/python%2Fassignment%20file%20handling%20C%20exception%20handling%20oopsitled9.ipynb

jupyter

assignment file handling, exception handling,oopsitled9

Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Trusted

JupyterLab Python 3 (ipykernel)

Successfully wrote content to SEENIYA24.
Welcome! File operation completed successfully.

[]: #9.Build a program to manage a university's course catalog. You want to define a base class Course that has the following properties:
course_code: a string representing the course code (e.g., "CS101")
course_name: a string representing the course name (e.g., "Introduction to Computer Science")
credit_hours: an integer representing the credit hours for the course (e.g., 3)
You also want to define two subclasses CoreCourse and ElectiveCourse, which inherit from the Course class.
CoreCourse should have an additional property required_for_major which is a boolean representing whether the course is required for a particular major.
ElectiveCourse should have an additional property elective_type which is a string representing the type of elective (e.g., "general", "technical", "liberal arts").

[19]: class Course:
 def __init__(self, course_code, course_name, credit_hours):
 self.course_code = course_code
 self.course_name = course_name
 self.credit_hours = credit_hours

 class CoreCourse(Course):
 def __init__(self, course_code, course_name, credit_hours, required_for_major):
 super().__init__(course_code, course_name, credit_hours)
 self.required_for_major = required_for_major

 class ElectiveCourse(Course):
 def __init__(self, course_code, course_name, credit_hours, elective_type):
 super().__init__(course_code, course_name, credit_hours)
 self.elective_type = elective_type

 # Example usage
 core_course = CoreCourse("CS101", "Introduction to Computer Science", 3, True)
 elective_course = ElectiveCourse("ART201", "Introduction to Painting", 2, "liberal arts")

 print(f"Core Course: {core_course.course_name}, Required for major: {core_course.required_for_major}")
 print(f"Elective Course: {elective_course.course_name}, Type: {elective_course.elective_type}")

Core Course: Introduction to Computer Science, Required for major: True

27°

ENG IN 10:10 PM 20-10-2024

Not syncing

Home

Home

assignment file handling, exce

GitHub - Search

SEENIYA24/Python-Modules: F

localhost:8888/notebooks/python%2Fassignment%20file%20handling%2C%20exception%20handling%2Coopsitled9.ipynb

Jupyter assignment file handling, exception handling,oopsitled9 Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Trusted

JupyterLab Python 3 (ipykernel)

```
print(f"Elective Course: {elective_course.course_name}, Type: {elective_course.elective_type}")

Core Course: Introduction to Computer Science, Required for major: True
Elective Course: Introduction to Painting, Type: liberal arts

[ ]: #10.Create a Python module named employee that contains a class Employee with attributes name, salary and methods get_name() and get_salary(). Write a pr

[ ]: class Employee:
    def __init__(self, name, salary):
        self._name = name
        self._salary = salary

    def get_name(self):
        return self._name

    def get_salary(self):
        return self._salary

from employee import Employee

def main():
    # Get employee information from user
    name = input("Enter employee name: ")
    salary = float(input("Enter employee salary: "))

    # Create an Employee object
    emp = Employee(name, salary)

    # Display employee information
    print("\nEmployee Information:")
    print(f"Name: {emp.get_name()}")
    print(f"Salary: ${emp.get_salary():.2f}")

if __name__ == "__main__":
    main()
```

27°

ENG IN 10:10 PM 20-10-2024