```python
import os
import requests
import shutil

from Logger import Logging

logger_obj = Logging('Advance Image Downloader')  # Creating a custom based logger
logger_obj.initialize_logger()  # Instantiating the logger object


class Download:

    def __init__(self, result=None):
        """
        This function initializes the content of the downloaded files
        :param result: Cassandra object of the downloadable result content
        """
        try:
            self.content = result
        except Exception as e:
            logger_obj.print_log('(Download.py(__init__) - Something went wrong ' + str(e), 'exception')
            raise Exception(e)

    @staticmethod
    def create_dir(req_id):
        """
        This function is used to create a directory for storing the images
        """
        try:
            # If the folder doesn't exist then create that folder inside the directory
            if not os.path.exists(req_id):
                os.mkdir(req_id)

        except Exception as e:
            logger_obj.print_log('(Download.py(create_dir) - Something went wrong ' + str(e), 'exception')
            raise Exception(e)

    def download_images(self, search_term, req_id):
        """
        This function is used to download the images over the internet and then store in the folder created
        """
        try:
            counter = 1
            for row in self.content:
                url = row.url
                req = requests.get(url, stream=True, headers={"User-Agent": "Mozilla/5.0 (X11; Linux x86_64) "
                                                              "AppleWebKit/537.36 (KHTML, like Gecko) "
                                                              "Chrome/51.0.2704.103 Safari/537.36"})
                print('Counter = {} Header = {} URL = {}'.format(counter, req.headers, url))

                logger_obj.print_log('Filetype is {}'.format(req.headers['Content-Type'].split('/')[1]), 'info')

                filetype = req.headers['Content-Type'].split('/')[1].split(';')[0]

                # For the files having webp as a extension
                if filetype not in ['jpeg', 'png']:
                    filetype = 'jpeg'

                req.raw.decode_content = True
                with open(req_id + '/' + search_term + '_' + str(counter) + '.' + filetype, 'wb') as file:
                    file.write(req.content)
                    file.close()
                    counter += 1

        except Exception as e:
            logger_obj.print_log('(Download.py(download_images) - Something went wrong ' + str(e), 'exception')
            raise Exception(e)

    @staticmethod
    def create_zip(req_id):
        """
        This function will be responsible for creating the zip file of the Downloaded Images folder
        :param req_id : The Unique id of the request
        """
        try:
            if not os.path.exists(req_id + '_zipfile.zip'):
                shutil.make_archive(req_id + '_zipfile', 'zip', req_id)

        except Exception as e:
            logger_obj.print_log('(Download.py(create_zip) - Something went wrong ' + str(e), 'exception')
            raise Exception(e)

    @staticmethod
    def delete_file(req_id):
        """
        This function will delete the given files
        :param req_id: The Unique id of the request
        """
        try:
            # If the zip exists then remove that from the system
            if os.path.exists(req_id + '_zipfile.zip'):
                os.remove(req_id + '_zipfile.zip')
                print('Zip file is deleted')

            # If the folder exists then remove that from the system
            if os.path.exists(req_id):
                shutil.rmtree(req_id)
                print('Image Folder is deleted')

        except Exception as e:
            logger_obj.print_log('(Download.py(delete_old_file) - Something went wrong ' + str(e), 'exception')
            raise Exception(e)

```