

```
# download_url.py
#
# ICS 32 Winter 2014
# Code Example
#
# This is a short program that demonstrates how to download the contents
# of a URL from the web and print it to the console. In the interest
# of simplicity, note that this program makes two assumptions:
#
# * The contents it downloads are text -- so, for example, I would expect
#   this program to crash if you tried to use it to download an image or
#   a video
# * The text is encoded using UTF-8 -- so, again, there may be some URLs
#   that will cause a crash because they've been encoded using something
#   else
#
# One thing that appears here that does not appear in the write-up
# accompanying this example is the use of the urllib.error.HTTPError
# exception, which is raised by urllib.request.urlopen() whenever an
# attempt to download a web page fails (e.g., because you try to
# download a page that does not exist). When that exception is raised,
# you can catch it and use its code attribute to determine what status
# code the server returned (e.g., 404 means that the page was not found,
# i.e., that it doesn't exist).
```

```
import http.client
import urllib.request
import urllib.error
```

```
def user_interface() -> None:
    while True:
        url = _choose_url()

        if len(url) == 0:
            return
        else:
            _download_and_print_url(url)
```

```
def _choose_url():
    print('Choose a URL to download (press Enter or Return to quit)')
    return input('URL: ').strip()
```

```
def _download_and_print_url(url: str) -> None:
    response = None

    try:
        response = urllib.request.urlopen(url)
        _print_url_contents(response)

    except urllib.error.HTTPError as e:
        print('Failed to download contents of URL')
        print('Status code: {}'.format(e.code))
        print()
```

```
finally:
    if response != None:
        response.close()

def _print_url_contents(response: http.client.HTTPResponse) -> None:
    content_bytes = response.read()
    content_string = content_bytes.decode(encoding='utf-8')
    content_lines = content_string.splitlines()

    print('Response contains {} line(s) of text'.format(len(content_lines)))
    print()

    for line in content_lines:
        print(line)

    print()
    print()

if __name__ == '__main__':
    user_interface()
```