.

13. Create of Debian packages.

(Multiple modules/code packaging of python).

Pull or take any FOSS project and one feature and then create the package on suitable OS

Compare RPM packaging with Debian packaging (on answer sheet)

For this example, we'll use **Beautiful Soup (bs4)**, a popular library for web scraping. We'll package a feature that extracts all hyperlinks from a web page using Beautiful Soup. Here's the full process:

Step 1: Set Up the Directory Structure

First, create the necessary folder structure for your Debian package

```
mkdir bs4-feature-package
cd bs4-feature-package
mkdir -p DEBIAN usr/local/lib/python3.x/dist-packages
```

Replace `3.x` with your actual Python version (e.g., `3.8`).

Step 2: Install Beautiful Soup (bs4) and Prepare the Feature Code

If Beautiful Soup is not already installed, install it:

```
sudo apt update
sudo apt install python3-pip
pip3 install beautifulsoup4
```

Step 3: Create a Python Script with a Specific Feature

Create a Python script called `bs4_feature.py` that uses Beautiful Soup to extract all hyperlinks from a given webpage.

Inside `bs4-feature-package/usr/local/lib/python3.x/dist-packages/`, create and open `bs4_feature.py`:

```
nano usr/local/lib/python3.x/dist-packages/bs4_feature.py
```

Add the following code to extract all hyperlinks from a webpage:

```
from bs4 import BeautifulSoup
import requests

def extract_hyperlinks(url):
    try:
        response = requests.get(url)
        if response.status_code == 200:
            soup = BeautifulSoup(response.text, 'html.parser')
            links = [a['href'] for a in soup.find_all('a',
href=True)]
            print("Hyperlinks found:")
            for link in links:
                print(link)
        else:
            print(f"Failed to retrieve the page. Status code:
{response.status_code}")
    except Exception as e:
        print(f"An error occurred: {e}")

# Test the function
if __name__ == "__main__":
    extract_hyperlinks("https://example.com")
```

Step 4: Create the Control File

Navigate to the `DEBIAN` directory and create a control file with the package metadata:

```
cd DEBIAN
nano control
```

Add the following content

```
Package: bs4-feature-package
Version: 1.0
Section: utils
Priority: optional
Architecture: all
Depends: python3, python3-bs4, python3-requests
Maintainer: Your Name <your.email@example.com>
```

```
Description: A feature package for Beautiful Soup to extract
hyperlinks from a webpage
```

Step 5: Build the Debian Package

Go back to the root directory of `bs4-feature-package`:

```
cd ..
```

Build the Debian package with:

```
dpkg-deb --build bs4-feature-package
```

This command creates `bs4-feature-package.deb` in the current directory.

Step 6: Install and Test the Package

Install the newly created Debian package:

```
sudo dpkg -i bs4-feature-package.deb
```

To verify it works, run the script:

```
python3 /usr/local/lib/python3.x/dist-packages/bs4_feature.py
```

If everything was set up correctly, it should output the hyperlinks found on https://example.com.

Hyperlinks found:

https://www.iana.org/domains/example

Like this

Comparison with RPM Packaging

Here's a quick comparison of Debian (DEB) and RPM packaging:

Aspect	Debian Packaging (DEB)	RPM Packaging
OS Compatibility	Debian-based (Ubuntu, Debian, etc.)	Red Hat-based (Fedora, CentOS)

Package Manager	dpkg (managed by apt)	rpm (managed by yum or dnf)
File Format	`.deb`	`.rpm`
Dependency Handling	Managed by `apt`	Managed by `yum` or `dnf`
Structure	Requires `DEBIAN/control` file	Requires `.spec` file
Build Tool	`dpkg-deb`	`rpmbuild`
Simplicity	Simple for small packages	More complex for larger packages
Popularity	Common on Ubuntu, Debian, etc.	Common on Red Hat, Fedora, etc.