

Lab 1: Integrating GPSS World, Python, and GitHub Copilot

Theory:

Simulation and modeling are essential techniques in various fields such as engineering, business, and computer science.

GPSS (General Purpose Simulation System) World is a simulation software package widely used for modeling and analyzing discrete event systems. This theoretical framework delves into the fundamental concepts and applications of GPSS World, providing insights into its theoretical underpinnings and practical uses in various domains.

Python is a versatile and widely used programming language known for its simplicity, readability, and vast ecosystem of libraries. GitHub Copilot, on the other hand, is an AI-powered code completion tool developed by GitHub and OpenAI, designed to assist developers in writing code more efficiently. This theoretical framework explores the foundational principles and practical implications of Python programming and GitHub Copilot in software development.

Installations:

GPSS World

1. Download the GPSS World installer from the official website.
2. Run the installer and follow the on-screen instructions.
3. Accept the license agreement and choose the installation directory.
4. Complete the installation process.
5. Launch GPSS World from the Start menu or desktop shortcut.

Python

1. Download Python from `python.org`.
2. Run the installer.
3. Follow installation prompts.
4. Verify installation with **`python --version`**.

GitHub Copilot

1. Sign up for the GitHub Student Developer Pack using your student credentials at `education.github.com/pack``.
2. Wait for your application to be approved. This usually takes a few days.
3. Once approved, access the GitHub Student Developer Pack dashboard.
4. Navigate through the available tools and services in the pack to find GitHub Copilot.
5. Follow the instructions provided to redeem GitHub Copilot.
6. After redeeming GitHub Copilot, you'll likely receive instructions on how to access and install it.

Integrating with VS Code

1. If you haven't already, download and install Visual Studio Code from `code.visualstudio.com`` and launch it.
2. Go to the Extensions view by clicking on the square icon on the sidebar or pressing `Ctrl+Shift+X``. Search for "GitHub Copilot" in the Extensions Marketplace and install the GitHub Copilot extension.
3. Sign in to your GitHub account within VS Code. This is necessary to authenticate your access to GitHub Copilot.
4. After installation, GitHub Copilot should be enabled automatically. You may need to reload VS Code for the changes to take effect.
5. After completion, GitHub Copilot is now integrated with VS Code! You can start using it by typing code in any supported programming language, and Copilot will provide code suggestions and completions as you write.

Now, let's build a program to randomly generate a number from 1 to 100 and also print our name and roll number.

Code:

```
```\n\nimport random\n\nrandom_number = random.randint(1, 100)\nprint("Generated Random Number: ", random_number)\nprint("Name: Mansij Maharjan\\nRoll Number: 11")\n```\n
```

OUTPUT:

```
mansijmaharjan@Mansijs-MacBook-Pro lab-1 % python3 random_number.py\nGenerated Random Number: 92\nName: Mansij Maharjan\nRoll Number: 11\nmansijmaharjan@Mansijs-MacBook-Pro lab-1 %
```

Let's use GitHub Copilot to do the same:

1. On VS Code, create a new Python file by clicking on the "File" menu, selecting "New File," and saving it with a **.py** extension (e.g., **random\_numbers.py**).
2. Use comments to describe what you want GitHub Copilot to generate. For example:  
# Generate a random integer between 1 and 100
3. Start typing code below the comment, such as **random\_number =** , and let GitHub Copilot suggest code based on the comment.
4. When GitHub Copilot provides suggestions, review them to ensure they match your intentions. If a suggestion meets your needs, accept it by pressing **Tab** or clicking on it.
5. Once you've accepted a suggestion, you can test the code by running the Python file. Right-click on the file in VS Code and select "Run Python File in Terminal" from the context menu.

Conclusion:

In conclusion, integrating GitHub Copilot with Visual Studio Code provides developers with a powerful tool for enhancing productivity and code quality. With Copilot's AI-powered code suggestions and completions, developers can write code more efficiently, explore new programming concepts, and reduce the time spent on routine tasks. By seamlessly integrating Copilot into their development workflow, developers can leverage its capabilities to streamline coding processes, collaborate more effectively, and build high-quality software with greater speed and precision.