AKTU B.Tech Second Year

PYTHON PROGRAMMING

Unit 5

Python file operations

LEC-3

Gateway classes

Today's Target

- **PYTHON WITH IDE**
- **AKTU PYQs**

By
Pragya Rajvanshi

Syllabus

Python packages: Simple programs using the built-in functions of packages matplotlib, numpy, pandas etc. GUI Programming: Tkinter introduction, Tkinter and PythonProgramming, Tk Widgets, Tkinter examples. Python programming with IDE.

Gateway classes

Python

- Python is a high-level, interpreted programming language known for its simplicity and readability
- You can use it to create websites, analyze data, automate tasks, or build apps.
- It's beginner-friendly and lets you write code that's easy to read and understand.

Features:

Easy to Learn and Use: Simple syntax like plain English.

Cross-Platform: Works on Windows, macOS, Linux, etc.

Large Library: Ready-to-use tools for tasks like math, web, and data.

Dynamic Typing: No need to define variable types explicitly.

Community Support: Lots of resources and help available online.

Gateway classes

How python is interpreted?

Python is interpreted because its code is executed line by line by the Python interpreter.

Here's how it works:

1.Source Code:

- You write Python code in a file (e.g., example.py).
- This is the human-readable version of your program.

2.Python Interpreter:

- When you run the code (e.g., using python example.py), the Python interpreter starts working.
- The interpreter reads the code line by line.

3.Bytecode Conversion:

- The interpreter translates your source code into an intermediate form called bytecode.
- Bytecode is a simplified version of your code that is easier for the computer to understand.
- This bytecode is stored temporarily in memory or as a .pyc file (if caching is enabled).

.

Gateway classes

4.Python Virtual Machine (PVM):

- The bytecode is executed by the Python Virtual Machine.
- The PVM takes each instruction from the bytecode and executes it on your computer.

5.Output:

- The interpreter runs the code step by step, producing results as it goes.
- If there's an error in your code, Python stops and shows an error message.

Static analysis

It is the process of analyzing code without executing it. It checks for bugs, errors, security vulnerabilities, and code quality issues by examining the source code or bytecode.

Key Points:

- **How it works:** Tools analyze the code structure, syntax, and logic to detect potential problems.
- **Purpose:** To catch errors early in the development process before the code is run or tested.
- **Examples:** Finding unused variables, type mismatches, or missing imports.

Gateway classes

Tools that help to find bugs or perform static analysis

- PyLint is a popular Python tool for analyzing code to ensure it adheres to coding standards, finds errors, and improves code quality.
- It performs static code analysis and provides suggestions for improvement.

Features of PyLint:

1.Code Quality Checks:

- Detects issues like unused variables, undefined variables, and unreachable code.
- Ensures your code follows the Python style guide (PEP 8).

2.Coding Standards:

- Highlights deviations from best practices.
- Enforces consistent naming conventions for variables, classes, and methods.

3. Error Detection:

- Finds common errors like missing imports, syntax issues, or potential bugs.

Gateway classes

Code Refactoring:

- Provides suggestions to refactor and simplify code to make it cleaner and more readable.

Custom Rules:

- Allows users to define custom rules for analysis or disable specific checks as needed.

- **PyChecker**
- It is a static analysis tool for Python that helps find bugs, errors, and inconsistencies in code.
- It was one of the earlier tools for Python linting, though it is less commonly used today compared to tools like PyLint.

Features:

1.Bug Detection:

- Identifies syntax errors and logical mistakes in code.
- Flags unused variables, modules, or imports.

2.Error Analysis:

- Detects type mismatches and missing function arguments.
- Highlights potential runtime errors.

Gateway classes

3. Code Warnings:

- Issues warnings for deprecated Python features or bad practices.

4.Cross-Version Compatibility:

- Checks for compatibility issues across different Python versions.

Python decorators

- A decorator is a special type of function in Python that allows you to modify or enhance the behavior of other functions or methods without changing their code.
- Decorators are often used to add functionality like logging, timing, or access control to functions.

Gateway classes

main.py

```python
def decorator_function(original_function):
    def wrapper_function():
        print("Before function execution")
        original_function()  # Call the original function
        print("After function execution")
    return wrapper_function

@decorator_function  # Apply the decorator
def greet():
    print("Hello, World!")

greet()
```

```
Before function execution
Hello, World!
After function execution
```

Gateway class

What is IDE?

- An IDE (Integrated Development Environment) is a software application that provides tools to help programmers write, debug, and manage code efficiently.
- It integrates several features into one interface, making the development process easier and faster.

1.Code Editor: A text editor with features like syntax highlighting, autocomplete, and code formatting.

2.Debugger:Helps identify and fix errors in your code by running it step by step.

3.Compiler/Interpreter: Converts your source code into executable code (or interprets it for languages like Python).

4.Project Management: Allows you to organize files and folders within a project.

5.Integrated Tools: Includes tools for version control (e.g., Git), testing, and deployment.

.

Gateway classes

SOME OF IDE

1. PyCharm

- It is a popular Integrated Development Environment (IDE) specifically designed for Python development.
- It is developed by JetBrains and provides many features to help developers write, debug, and manage Python projects efficiently.

Key features :

1.Code Editor:

- Smart code completion and suggestions.
- Syntax highlighting and formatting.

2.Debugger:

- Powerful tools to identify and fix errors in Python code.

3.Code Analysis:

- Highlights coding issues and suggests improvements.

4.Project Management:

- Helps organize files and dependencies in large projects.

5.Support for Web Development:

- Includes frameworks like Django and Flask.

Gateway classes

6.Testing Tools:

- Includes support for running unit tests directly from the IDE.

Why Use PyCharm?

- Simplifies Python development with smart tools.
- Boosts productivity with features like code analysis and debugging.
- Ideal for both beginners and experienced developers.

2. VS Code

- VS Code (Visual Studio Code) is a lightweight, powerful, and free code editor developed by Microsoft.
- It is widely used for writing, debugging, and managing code in multiple programming languages, including Python, JavaScript, C++ etc.

Why Use VS Code?

- **Free:** Fully free and open-source.
- **Lightweight:** Ideal for both small scripts and large projects.
- **Customizable**: Extensions let you tailor it to your needs.
- **Community:** Huge user base with plenty of tutorials and resources.

Gateway classes

Key Features of VS Code:

1.Lightweight and Fast:

Designed to be minimalistic and run efficiently on most systems.

2.Code Editor:

Syntax highlighting, autocompletion, and IntelliSense (smart code suggestions)

3.Debugger:

Debug code with breakpoints, variable inspection, and call stack analysis.

4.Cross-Platform:

Available for Windows, macOS, and Linux.

5.Customizable:

Personalize the editor with themes, fonts, and keyboard shortcuts.

6.Multi-Language Support:

Supports many languages (Python, Java, JavaScript, C#, etc.) with extensions.

Gateway classes

3.Eclipse

- It is a powerful, open-source Integrated Development Environment (IDE) primarily used for developing Java applications.
- It is also versatile and supports other programming languages.

Key Features of Eclipse:

1.Java-Focused:

- Built specifically for Java development with tools like auto-completion, debugging, and refactoring.

2.Project Management:

- Organize large projects easily with a structured interface.

3.Integrated Debugger:

- Set breakpoints, inspect variables, and step through your code for debugging.

4.Cross-Platform:

- Available for Windows, macOS, and Linux.

5.Version Control:

- Built-in support for Git and other version control systems.

Gateway classes

4.Spyder

- **Spyder (Scientific Python Development Environment) is an open-source Integrated Development Environment (IDE) specifically designed for data science and scientific computing in Python.**
- **It is widely used for tasks like data analysis, machine learning, and visualization.**

Key features of spyder

1.Data Visualization:

- **Supports inline plotting using libraries like Matplotlib and Seaborn.**

2.Integrated Debugger:

- **Step through your code to debug and fix errors.**

3.Project Management:

- **Organize multiple scripts and files in a structured manner.**

4.Code Analysis:

- **Includes tools like Pylint to detect errors and enforce coding standards.**

5.Support for Scientific Libraries:

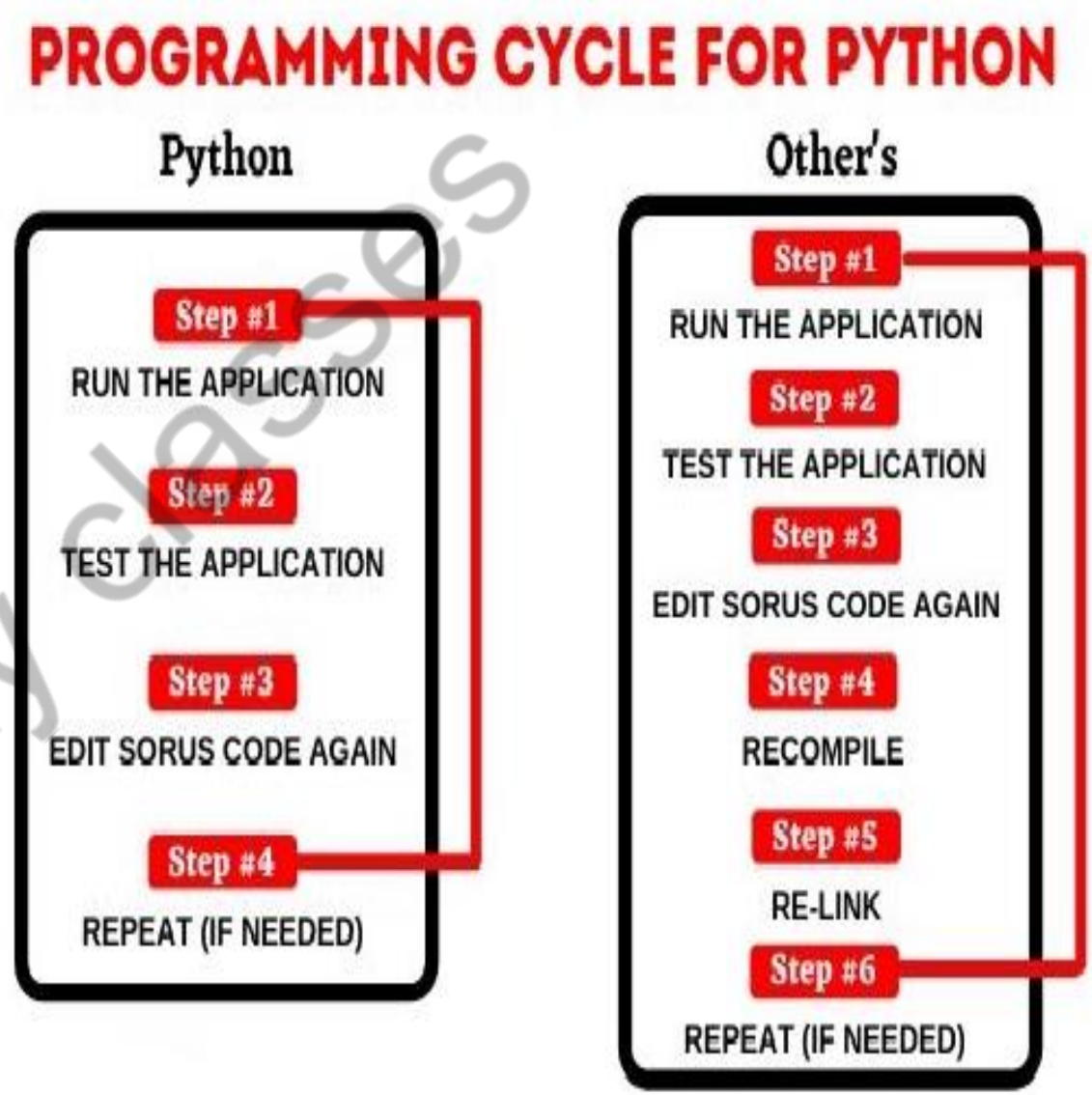
- **Seamless integration with NumPy, Pandas, SciPy, and Matplotlib.**

Gateway classes

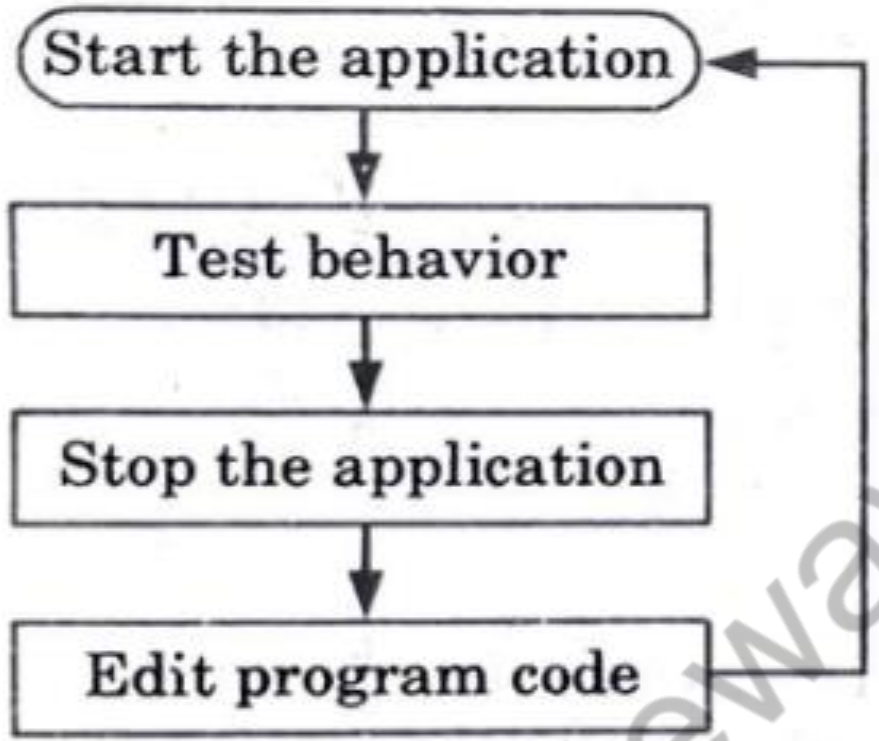
Programming Cycle for python

- The development cycle of Python is considerably shorter than that of traditional tools.
- There is no compilation or linking steps in Python.
- Python programs simply import modules at runtime and use the objects they contain. Because of this, Python programs run immediately after changes are made.

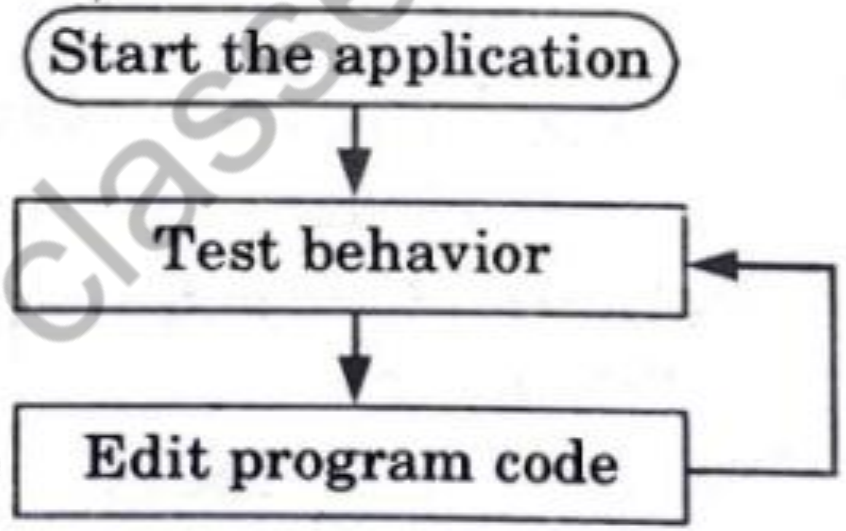
.



Gateway classes



a) Python's programming cycle



(b) Python's programming cycle with module reloading

Gateway classes

What is PEP 8?

- **PEP 8 (Python Enhancement Proposal 8) is the official style guide for Python code.**
- **It provides guidelines and best practices to help developers write clean, readable, and consistent Python code.**

Why Follow PEP 8?

- **Readability**: Makes code easier to read and understand.
- **Consistency**: Helps teams maintain a uniform coding style.
- **Best Practices**: Encourages efficient code.

Key Guidelines of PEP 8:

1.Indentation:

Use **4 spaces per indentation level**.

2. Line length

PEP 8 recommends limiting each line of code to a maximum of **79 characters** and comments or docstrings to **72 characters.**

Gateway classes

Spaces

- Use spaces around operators and after commas.
- Don't add spaces inside parentheses or brackets.

Imports

- Put imports at the top of the file.
- Import one library per line.

Comments

- Write meaningful comments to explain *why* something is done.
- Use # and keep comments short and clear python.

Functions

- Keep your functions small and focused on one
- Name them clearly to show what they do.

Gateway classes

AKTU QUESTIONS		
Q.1	What is python? How python is interpreted? What are the tools that help to find bugs or perform static analysis? What are python decorators?	AKTU 2019-20 AKTU 2020-21 AKTU 2022-23
Q.2	Describe how to generate random number using numpy. write a python m program to create an array of 5 random number integer between 10 and 50	AKTU 2023-24
Q.3	Explain the dataFrame in pandas. Write a python program to create an dataFrame from dictionary and print it	AKTU 2023-24

Gateway classes


Thank
you