

Report of Certification Course on “Unboxing OAPI Feature of the Civil (Structural) Engineering Software using Python Programming”

Course Overview: The School of Engineering and Technology (SET) at Gujarat Technological University (GTU) successfully conducted a certification course titled "Unboxing OAPI Feature of the Civil (Structural) Engineering Software using Python Programming." The course was designed to provide participants with essential programming skills in Python tailored for civil and structural engineering applications. It included theoretical learning and practical sessions covering topics such as Monte-Carlo simulations, data handling with Pandas and NumPy, visualization techniques, and the integration of Python with industry-standard software like SAP2000, AutoCAD, and ETABs.

Course Details:

- **Duration:** 55 hours
- **Mode of Delivery:** Online
- **Participants:** Undergraduate and postgraduate students, PhD scholars, and industry professionals.

Key Highlights:

- **Inaugural Ceremony:** The course began with an inaugural ceremony featuring addresses from key university figures and industry experts, setting the stage for an insightful learning experience.
- **Expert Talks:** The course included expert talks by industry leaders, providing participants with advanced insights into Python's applications in structural engineering and future trends in civil engineering.
- **Hands-on Sessions:** The course emphasized practical applications, with hands-on sessions that allowed participants to apply Python programming to real-world engineering problems.

Outcomes:

- Participants gained proficiency in Python, enabling them to automate engineering tasks, perform data analysis, and integrate Python with civil engineering software.
- The course equipped participants with skills to conduct advanced simulations and develop innovative solutions to engineering challenges.
- The program concluded with a valedictory ceremony, recognizing the achievements of participants and encouraging further learning.

Feedback and Recommendations:

- **Overall Satisfaction:** Participants rated the course highly, with excellent feedback on content quality, clarity of presentation, and the effectiveness of hands-on sessions.
- **Recommendations for Future Courses:** Suggestions included conducting more Faculty Development Program (FDP) sessions and providing additional hands-on practice.

Conclusion: The certification course was a resounding success, meeting its objectives and providing participants with valuable skills that will enhance their professional development in the field of civil engineering. The positive feedback underscores the course's impact and highlights the demand for similar programs in the future.

Prepared by:

- Prof. (Dr.) Kaushik Gondaliya, Assistant Professor
- Prof. Komal Prajapati, Assistant Professor
- Prof. (Dr.) Bhavdip Moghariya, Assistant Professor

Glimpse of the Event

The image is a collage of screenshots from a virtual event. At the top, a Zoom meeting window shows several participants: Dr. J. A. Amin, Twinsky N. Paisan, Mohd Sameer, and Khudai Birdi Nazari. The main part of the collage is a software interface. On the left, a code editor displays the following code:

```

flowchart TD
    A[ ] --> B[flowchart.node("E", "Define Load Combinations")]
    B --> C[flowchart.node("F", "Assign Floor Diaphragm and Meshes")]
    C --> D[flowchart.node("G", "Assign Fixed Supports at Base")]
    D --> E[flowchart.node("H", "Run Analysis")]
    E --> F[flowchart.node("I", "Run Design Check")]
    F --> G[flowchart.node("J", "Optimization Loop")]
    G --> H[flowchart.node("K", "Assign Initial Column Dimensions")]
    H --> I[flowchart.node("L", "Run Design Check")]
    I --> J[flowchart.node("M", "Design Passes?")]
    J --> K[flowchart.node("N", "Increase Dimensions")]
    K --> L[flowchart.node("O", "Next Floor Pair")]
    L --> M[flowchart.node("End", "End")]

    # Add edges
    flowchart.edges["SA", "AC", "BC", "CD", "DE", "EF", "FG", "GH", "HI", "JK"]
    flowchart.edge("K", "N", label="no")
    flowchart.edge("N", "O", label="yes")
    flowchart.edge("O", "J")

    # Render the graph
    flowchart.render("etabs_model_workflow", format="png", view=true)

    etabs_model_workflow.png

    [ ] free graphviz import Digraph

    # Create a new directed graph
    flowchart = Digraph(comment="ETABS Model workflow", format="png")
    
```

On the right side of the software interface, a flowchart titled 'etabs_model_workflow_colorful_high_quality.png' is displayed. It starts with a 'No Input' node, followed by a 'No Input' node, then a 'No Input' node, and a 'No Input' node. The flowchart includes decision points for 'Design Passes?' and 'Increase Dimensions?'. The process ends with 'End of Flowchart'.

At the bottom, a window shows a 3D model of a building structure, likely generated from the software.