

KCG Programming Club – Odd Semester

KCG Programming Club (Session I)

Venue: Billgates Lab (Main building)

Date:27.09.25

INTRODUCTION TO PYTHON

The **Introduction to Python** workshop was organized to provide participants with a basic understanding of Python programming. The session focused on key concepts such as **variables, data types, conditional statements, loops, and functions**. Participants were also introduced to common coding tools and simple program development using Python.

The session began with an introduction to Python and its importance in the field of programming and software development. Participants were familiarized with the basic structure of a Python program and were guided through the fundamental concepts required for beginning their programming journey. The session emphasized understanding the purpose of variables and data types and how they are used to represent and process information in a program.

The participants were also introduced to **conditional statements**, which are used to make decisions based on specific conditions. Through simple programming examples, learners were able to understand how conditions can control the flow of execution. The session further introduced **loops**, helping participants understand how repetitive tasks can be performed efficiently without writing the same statements multiple times.

Another important concept covered during the session was **functions**. Participants learned the basic purpose of functions and how they can be used to organize a program into smaller and reusable sections. This helped them understand the importance of writing structured and manageable programs.

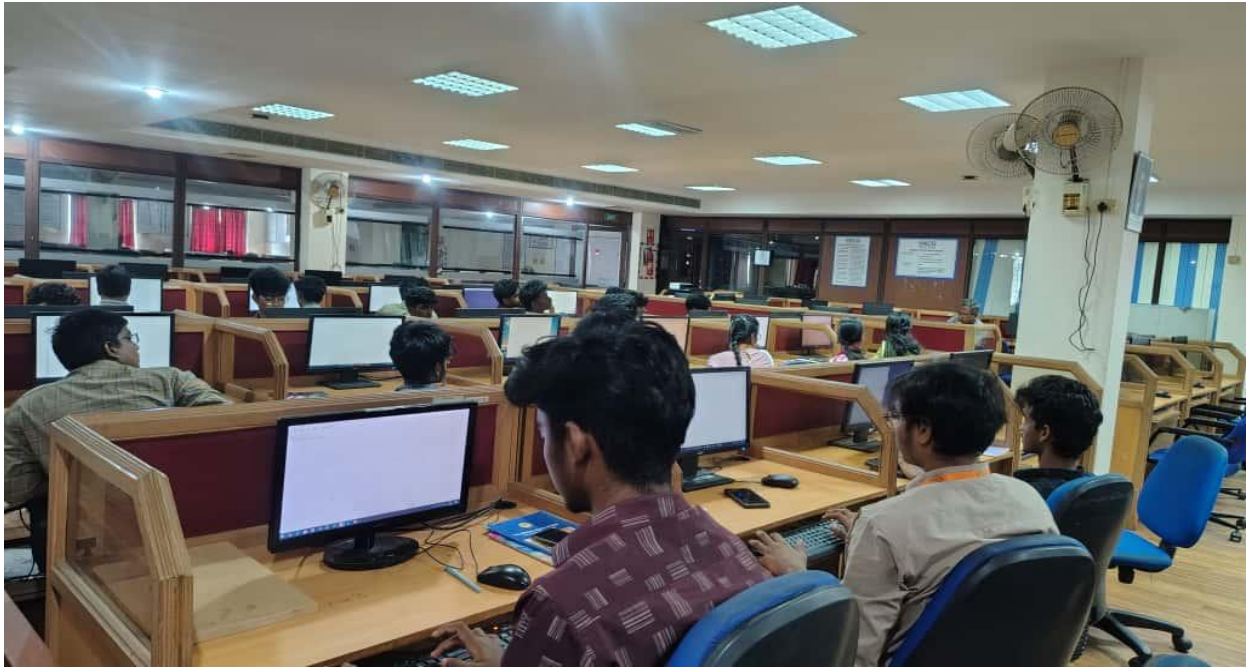
Participants were also introduced to common coding tools and simple program development using Python. The hands-on approach encouraged participants to write and execute basic Python programs and observe the output. This

practical exposure helped them connect the theoretical concepts discussed during the session with actual programming implementation.



The main aim of the workshop was to help learners improve their logical thinking and programming skills through hands-on practice. To make the session interactive, a **Python Quiz** was conducted, allowing participants to test their knowledge and strengthen their understanding of the topics covered.

The quiz provided an opportunity for participants to recall the concepts introduced during the session and apply their understanding in an interactive manner. It also encouraged active participation and created an engaging learning environment.



Overall, the workshop offered a practical learning experience and encouraged participants to explore Python as a powerful and beginner-friendly programming language. The session highlighted the importance of writing clear, structured, and reusable code for effective program development.

The session successfully established a foundation for the subsequent programming club sessions. By introducing the basic building blocks of Python and combining conceptual learning with practical exercises and a quiz, the workshop helped participants develop confidence in approaching programming problems.

KCG Programming Club (Session II)

Venue: Billgates Lab (Main building)

Date:11.10.25

ADVANCED PYTHON CONCEPTS AND MINI PROJECT EXPLORATION

The second session of the Python workshop focused on strengthening core programming skills with topics such as **lists, arrays, string handling, and simple pattern problems**. Participants learned how to store, access, and manage data using Python lists and arrays, along with performing essential string operations like concatenation, slicing, searching, and formatting.

The session served as a continuation of the concepts introduced in the first session. Participants were encouraged to build upon their existing knowledge and understand how Python can be used to work with different types of data efficiently.

The concepts of **lists and arrays** were introduced with emphasis on storing and managing collections of data. Participants learned how elements can be accessed and managed, helping them understand the importance of data organization in programming.

The session also provided an introduction to **string handling**. Participants explored essential operations such as concatenation, slicing, searching, and formatting. These concepts helped learners understand how textual information can be processed and manipulated using Python.



The session also included practice on basic pattern printing, helping learners improve logical thinking and loop control. In addition to new concepts, a quick recap of previous topics was conducted to connect earlier lessons with the current session.

Pattern problems provided participants with an opportunity to apply loops and logical thinking to generate specific outputs. This activity helped learners understand how programming logic can be converted into step-by-step solutions.

The recap of previous topics was particularly useful in connecting the fundamentals of Python with the concepts introduced in the current session. Participants were encouraged to recall variables, conditions, loops, and functions before applying them to more practical problems.



To encourage creativity and practical implementation, a **mini project idea** was introduced:

□ **CALCULATOR USING FUNCTIONS** – performing addition, subtraction, multiplication, and division with user-defined functions.

The calculator project introduced participants to the practical use of functions. By separating different arithmetic operations into user-defined functions, learners could understand how a simple application can be divided into reusable components.

Another activity that combined all learned topics:

□ **NUMBER GUESSING GAME** – integrating lists/arrays, loops, and string messages for an interactive experience.

The number guessing game provided a more interactive programming experience. It encouraged participants to combine multiple concepts and understand how individual programming elements can work together to create a complete activity.



Overall, Session 2 helped participants enhance problem-solving skills and apply Python concepts effectively through hands-on exercises and mini-project exploration.

The session strengthened the participants' ability to work with data, manipulate strings, apply loops, and develop simple logic-based applications. The introduction of mini-project ideas also encouraged learners to move beyond individual programming concepts and think about how these concepts can be combined to create functional programs.

KCG Programming Club (Session III)

Venue: Charles babbage Lab

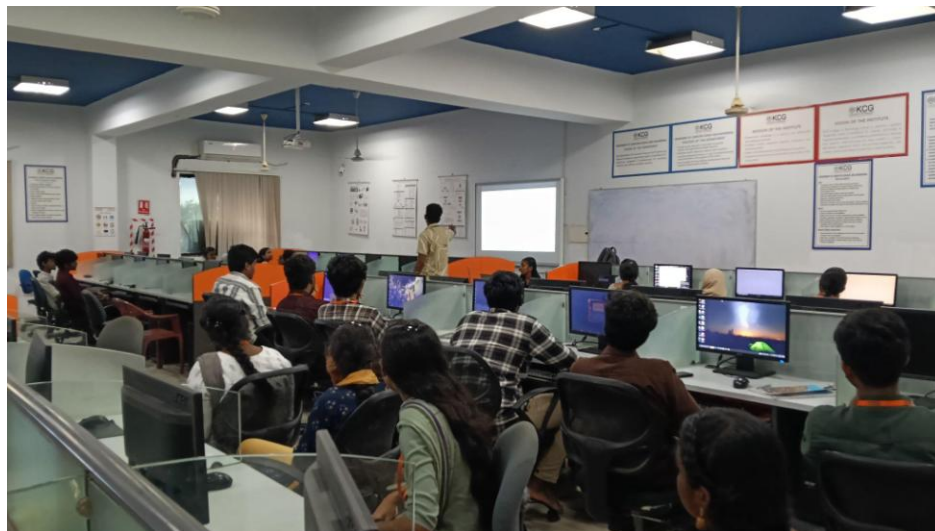
Date: 25.10.25

FILE HANDLING, RECURSION & OOPS BASICS

Session 3 focused on advancing participants' Python skills with **file handling, recursion, basic OOPS, and a practical game activity—Tic-Tac-Toe**. Learners were introduced to reading and writing files to store and retrieve data, an essential feature for many real-world applications.

The session introduced participants to the concept of **file handling**, which is an important part of programming when information needs to be stored and retrieved. Participants learned the basic idea behind reading data from files and writing information into files, thereby understanding how programs can work with stored information.

The concept of file handling provided learners with an understanding of how programming can extend beyond temporary data processing and interact with external files.



The concept of recursion was explained through simple examples to demonstrate how a function can call itself to solve problems like factorial calculation or number series. Participants were introduced to the idea of breaking a problem into smaller versions of the same problem and using a function repeatedly to obtain the required result.

Recursion helped participants strengthen their logical thinking and understand another important programming approach for solving problems.

Participants also explored Object-Oriented Programming basics, including classes, objects, and methods, highlighting how OOPS helps in building clean, organized, and reusable code.

The introduction to **classes, objects, and methods** provided participants with a foundation for understanding Object-Oriented Programming. These concepts helped learners understand how related data and functionality can be organized together, making programs easier to manage and reuse.

To reinforce the topics, a Tic-Tac-Toe game was implemented using functions, loops, and conditional logic. This activity helped participants practice logical thinking, input validation, and step-by-step game flow design.

The Tic-Tac-Toe activity provided an opportunity to combine previously learned Python concepts with the concepts introduced in the session. Participants could understand how functions, loops, conditions, and user inputs can work together to develop an interactive program.

The activity also helped learners understand the importance of input validation and proper control of the game flow. By working through the different stages of the game, participants gained practical exposure to problem-solving and application development.

Overall, Session 3 strengthened programming confidence and prepared participants for more complex Python projects.

The session provided participants with exposure to important Python concepts that are useful in real-world programming. The combination of file handling, recursion, OOPS basics, and the Tic-Tac-Toe activity created a practical learning environment and encouraged participants to continue exploring Python development.



Faculty Coordinators

Mrs. Priyanka G, Assistant professor / CSE

Student Coordinators

Kanagapriya P –III YEAR CSE

Sangamithra D –III YEAR CSE

KCG Programming Club

Even Semester

KCG Programming Club (Session I)

Venue: Charles babbage Lab

Date : 28.2.26

**PYTHON MODULES, EXCEPTION HANDLING
& HACKATHON INSIGHTS**

Session 4 began with a recap of Python basics, reinforcing the concepts covered in the previous sessions and helping participants strengthen their understanding before moving on to advanced topics. The recap provided continuity between the earlier sessions and the advanced concepts introduced in the current session.

The session then introduced modules and libraries, explaining how Python's built-in and external libraries can be used to simplify programming tasks, improve code reusability, and enhance development efficiency.

Participants were introduced to the importance of **modules and libraries** in Python programming. They learned how existing functionality can be utilized to simplify development and avoid unnecessary repetition of code. This helped participants understand how reusable resources can improve programming efficiency and support the development of larger applications.

Participants also explored exception handling, learning how to manage runtime errors using `try`, `except`, `else`, and `finally` blocks to build more reliable and user-friendly programs.

The discussion on exception handling helped participants understand the importance of handling unexpected situations during program execution. The use of `try`, `except`, `else`, and `finally` blocks provided learners with a structured approach to managing runtime errors and improving the reliability of Python programs.

In addition, the session covered functional programming tools in Python, introducing useful built-in functions and concepts that enable developers to write cleaner, more efficient, and modular code.

Apart from technical concepts, the session included an informative discussion on how to approach hackathons, with guidance on problem identification, teamwork, idea validation, project planning, and effective presentation strategies.

The hackathon discussion helped participants understand that successful participation in technical competitions involves more than programming. Participants were guided on identifying meaningful problems, validating ideas, working effectively as a team, planning the project, and presenting the final solution clearly.

Participants were also introduced to the different domains in Computer Science and Engineering (CSE), such as Artificial Intelligence, Data Science, Cybersecurity, Cloud Computing, Web Development, and Software Engineering, helping them understand various career opportunities and choose areas based on their interests.

The introduction to different CSE domains helped participants gain awareness of the wide range of opportunities available within computer science. It encouraged learners to explore different technical areas and identify domains that match their interests and future learning goals.

The session concluded with an interactive quiz, where participants tested their understanding of both the recap and newly covered topics.



Overall, Session 4 strengthened programming fundamentals, introduced advanced Python concepts, and provided valuable insights into technical competitions and career pathways in the field of computer science.

The session successfully combined technical learning with career-oriented guidance. The combination of Python modules, exception handling, functional programming concepts, hackathon insights, CSE domains, and the interactive quiz provided participants with both programming knowledge and broader awareness of opportunities in the field.

KCG Programming Club (Session II)

Venue: T-63

Date: 14.03.26

FULL-STACK DEVELOPMENT WITH GIT & REACT

Session 5 introduced participants to the fundamentals of **Git and GitHub**, emphasizing the importance of version control, collaboration, and project management in software development.

The session provided participants with an understanding of how version control systems support software development. Git and GitHub were introduced as important tools for managing project changes, collaborating with other developers, and maintaining project history.

The session provided an overview of **Full-Stack Development**, explaining how the frontend, backend, and database work together to build complete web applications.

Participants were introduced to the overall structure of a full-stack application. The interaction between the frontend, backend, and database was explained to help learners understand how different layers of a web application work together.



Participants explored **Frontend Development using React.js**, learning the basics of components, JSX, and creating interactive user interfaces.

The introduction to React.js helped participants understand component-based frontend development. The concepts of components and JSX provided learners with a foundation for creating structured and interactive user interfaces.

They were also introduced to **Backend Development using Node.js**, understanding how server-side applications process requests and communicate with databases.

The backend discussion helped participants understand the role of server-side development in a web application. Participants were introduced to how backend applications process requests and communicate with databases to provide the required data and functionality.

The session demonstrated **frontend and backend integration**, allowing participants to see how data flows between the client and the server.

This integration demonstrated the relationship between the different parts of a full-stack application and helped learners understand how a frontend application can communicate with a backend service to exchange information.



In addition, learners gained an introduction to **database fundamentals**, **NPM (Node Package Manager)**, and **package installation**, helping them understand how dependencies are managed in modern development environments.

The discussion on database fundamentals provided participants with a basic understanding of the role of databases in storing and managing application data. NPM and package installation were introduced to help learners understand how external packages and dependencies can be added and managed during project development.

The session concluded with a **small project demonstration** that showcased the complete development workflow—from setting up a project and installing packages to integrating the frontend, backend, and database.

The project demonstration helped participants visualize the complete development process and understand how the concepts discussed throughout the session can be combined to create a functional web application.

Overall, Session 5 provided participants with a practical understanding of full-stack web development and the tools commonly used in real-world software projects.

The session gave participants exposure to modern development tools and technologies, including Git, GitHub, React.js, Node.js, databases, and NPM. It also encouraged participants to understand how different technologies work together as part of a complete software development workflow.

KCG Programming Club (Session III)

Venue: T-63

Date: 28.3.26

CODING MARATHON

The final session of the Python workshop was conducted as a **Coding Marathon**, providing participants with an opportunity to apply the concepts learned throughout the workshop in a practical and competitive environment.

The Coding Marathon served as an activity where participants could apply the programming knowledge gained during the previous sessions. Rather than focusing only on individual concepts, the activity encouraged learners to combine their understanding and develop solutions to programming challenges.

The marathon included a variety of programming challenges covering topics such as **Python fundamentals, functions, lists, strings, file handling, recursion, exception handling, and Object-Oriented Programming (OOPS)**.

The wide range of topics provided participants with an opportunity to revise and apply the concepts covered throughout the workshop. The challenges encouraged participants to recall fundamental programming concepts as well as the advanced topics introduced during the previous sessions.



Participants were encouraged to analyze problems, develop efficient solutions, and implement them within the given time.

The activity emphasized the importance of understanding a problem before beginning implementation. Participants were encouraged to analyze the given requirements, develop logical solutions, and implement their ideas effectively within the available time.

The event promoted logical thinking, coding efficiency, and problem-solving skills while fostering healthy competition and teamwork among participants.

The competitive environment encouraged participants to improve their coding approach while also learning from the problem-solving process. The activity provided an opportunity to develop confidence in handling programming challenges under time constraints.



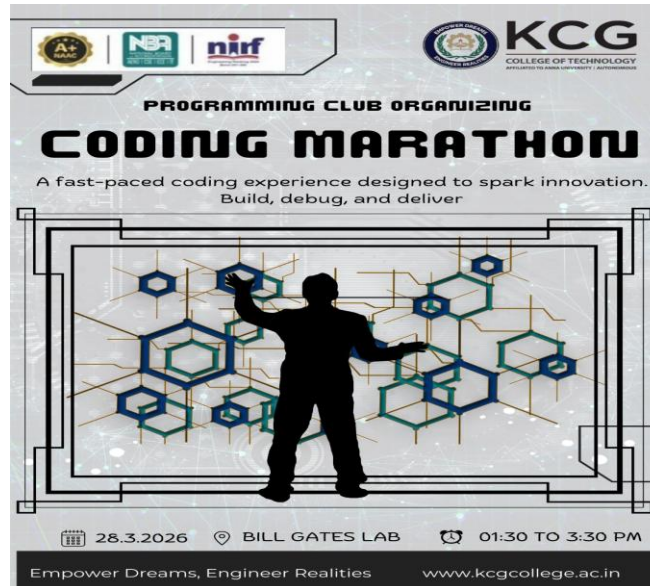
The Coding Marathon served as a comprehensive assessment of the knowledge gained during the workshop, allowing participants to showcase their programming abilities and build confidence in tackling real-world coding challenges.

The marathon provided an opportunity for participants to demonstrate their understanding of Python and their ability to apply different programming concepts to solve problems. It also helped them identify areas where they could further improve their programming and problem-solving abilities.



Overall, the final session successfully concluded the workshop by reinforcing practical learning and inspiring participants to continue their journey in Python programming.

The Coding Marathon provided a suitable concluding activity for the workshop by bringing together the concepts learned across the sessions. It encouraged participants to continue practicing programming, exploring new concepts, and developing their skills through practical implementation.



Faculty Coordinators

Mrs. Priyanka G, Assistant professor / CSE

Student Coordinators

Kanagapriya P –III YEAR CSE

Sangamithra D –III YEAR CSE