

PRODUCING TECH LEADERS FOR FUTURE



VISION MISSION & VALUES

Our Vision

To establish a state of the art global online coding school for School kids to catch up with the tech industry quickly



Our Mission

To excel the coding, mathematical and problem solving skills in school kids to explore their hidden talent through advanced programming technologies

Our Values

We believe to inculcate the following core values in our future tech leaders

01

SELF EFFICACY

We generate self-belief in the kids to dig out their hidden abilities to perform any task with confidence to achieve their goals.

02

SEEKING FOR LEARNING

We value inquisitiveness and growth of kids with different learning needs. We encourage them to become creative, logical thinkers and problem solvers for themselves and the society.

03

LEADERSHIP

Our teeny coders are the leader of the digital future. We enlighten them with individual and teamwork abilities, coupled with moral and ethical values, to serve the community.

04

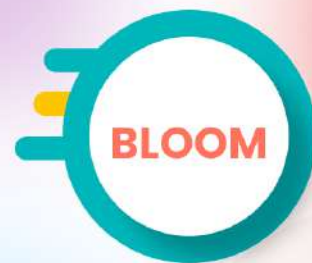
INCULCATION OF SKILLS

Every day, we are moving towards digitalization. We believe in inculcating coding, mathematical and problem solving skills in kids through our quality curriculum to meet the needs of the digital future.

WHY TEENY CODERS?

BLOOM'S TAXONOMY

We believe that every Teeny Coder is the leader of digital future. Our quality curriculum is designed based on these six levels (Create, Evaluate, Analyse, Apply, Understand and Remember) for effective learning. Teeny Coders have 0% compromise policy towards quality education, and adopt the standard guidelines.



FACE MODEL

Each teeny coder's learning matters. Therefore, we have developed our own FACE **FUN-TO-LEARN, ADVANCED, CREATIVE AND EVOLVING** model to verify that our curriculum is nourishing every teeny coder.



QUALITY CURRICULUM

Our Quality Curriculum is one of our main products. Our Fun-to-Learn, Advanced, Creative, and Evolving Curriculum is based on Bloom's Taxonomy Standards, which makes sure that every Teeny Coder is obtaining the best coding, problem solving, and cognitive skills.



COMPETENT FACULTY

We have selected the best faculty for our Teeny Coders, who are graduates from renowned universities with great teaching experience at academia and industry levels. Our faculty is energetic, efficient, and passionate to teach our digital future leaders.



VARIETY OF COURSES

We, at TEENY CODERS, offer a variety of flavours (courses) which are specifically designed for grade 1 to grade 12 kids. Every course comprises of three difficulty levels (Beginner, Intermediate and Expert). We make sure that every TEENY CODER enjoys their code learning journey with solid concepts.



STEERING LEADERSHIP

Teeny Coders leadership has combined experience of more than 25 years in academia and industry. Therefore, every teeny coder's future is bright and safe because our leadership knows what is best for your kids.

PYTHON PROGRAMMING CURRICULUM



INTERMEDIATE LEVEL



Course Contents

20 Lectures • 24 Activities • Duration: 2-3 Months



| LECTURE NO. | TOPICS : ACTIVITIES |
|-------------|--|
| Lecture 1 | ● Emendation of some concepts : Emendation of Previous Level Concepts |
| Lecture 2 | ● Proceed Functions : Conversion of Binary Value |
| Lecture 3 | ● Proceed Functions : Mad Libs |
| Lecture 4 | ● Practice of Functions : Solving Problems |
| Lecture 5 | ● Exception Handling : Catching Exceptions |
| Lecture 6 | ● Exception Handling : Try with Else Clause |
| Lecture 7 | ● File Handling (Text File) : Operations on Text File |
| Lecture 8 | ● File Handling (CSV File) : Operations on CSV File |
| Lecture 9 | ● Practice of File Handling : Crayola Crayons |
| Lecture 10 | ● Concept of OOP : Classes, Methods, Objects |
| Lecture 11 | ● Concept of OOP (Inheritance) : Methods in Inheritance |
| Lecture 12 | ● Concept of OOP (Polymorphism) : Functions & Polymorphism in Class Method |
| Lecture 13 | ● Concept of OOP (Abstraction) : Abstract Base Class Methods |
| Lecture 14 | ● Concept of OOP (Encapsulation) : Hiding Members |
| Lecture 15 | ● Modules : Creating, Using & Operations on Built -in Modules |
| Lecture 16 | ● Scripting (Scripting Modules) : JSON Encoded Data to Objects |
| Lecture 17 | ● Fun Coding Problems : Solving Coding Problems |
| Lecture 18 | ● Fun Coding Problems : Solving Coding Problems |
| Lecture 19 | ● Project : Fixing a Skeleton Code |
| Lecture 20 | ● Project : Fixing a Skeleton Code |