1st round syllabus

This syllabus does not encompass the entire IOAI syllabus. It has been purposefully shortened to make the 1st round easier and more approachable. Also be mindful that the included syllabus is a subset of the 2026 IOAI syllabus and does not wander outside its domain. The IOAI 2027 syllabus is still not out yet, but we can expect it to be very similar and analogous to the 2026 syllabus.

FULL IOAI 2026 syllabus:
https://ioai-official.org/wp-content/uploads/2025/10/Syllabus.pdf

Note: The topics listed here may seem a lot, but in reality there are very few as you only need to learn the core foundational stuff. In addition we have also listed explicitly what you need to know. The indepth explanation makes the syllabus look long, but in reality it is quite short and can be easily covered.

Topic #1: Python, its libraries and CPU vs GPU

Python Basics (Loops, Functions, etc.)	Students should show comprehensive understanding of python, they should be able to understand how to use functions, classes, loops, OOPs and other fundamental python and programming concepts.
NumPy and Pandas for Data Handling	Students should know how to use the NumPy and Pandas library and manipulate dataframes, arrays, numpy arrays etc.
Scikit-learn for ML	They should show basic understanding of the scikit-learn library. They need not memorise every import and code, but should show understanding about what this library is and what it is used for. (theory only)
Training Models on CPU and GPU	Students should be able to differentiate between when to use CPU and when to use GPU, what are the pros and cons of each, and how tensors are more suited to GPU.

Topic#2: Classical Machine Learning

Introduction to Machine Learning	
Supervised Learning	Students should understand what supervised learning is and when it is used. They should understand the difference between

	classification and regression and be able to identify examples of each.
Linear Regression	Students should understand the basic idea of fitting a line/model to predict a continuous numerical value. They should understand features, coefficients/weights, predictions, and the general idea of minimizing prediction error. Students should have an intuitive understanding of the regression equation. Students should not focus heavily on mathematics.
Logistic Regression	Students should understand that logistic regression is primarily used for classification. They should understand the role of probabilities and the sigmoid function and how a probability can be converted into a class prediction.
L1 and L2 Regularization	Students should understand why regularization is used and how it helps reduce overfitting. They should understand the conceptual difference between L1 and L2 regularization and their effect on model parameters. Detailed mathematical derivations are not required.
K-Nearest Neighbors (K-NN)	Students should understand how K-NN makes predictions based on nearby data points. They should understand the role of K, distance/similarity, and the effect of choosing a very small or very large K.
Decision Trees	Students should understand how decision trees split data using features and decision rules. They should understand nodes, branches, leaves, depth, and the basic idea behind choosing a split. Students should be able to interpret a simple decision tree.
Model Ensembles	Students should understand why combining multiple models can improve performance. They should know the basic ideas behind bagging, random forests, and gradient boosting, and understand the conceptual difference between them. Detailed implementation is not required.
Support Vector Machines (SVM)	Students should understand the basic goal of an SVM, including separating classes using a decision boundary and maximizing the margin. They should have a basic conceptual understanding of support vectors and kernels.
Unsupervised Learning	Students should understand what unsupervised learning is and how it differs from supervised

	learning. They should understand that the model attempts to discover patterns or structure in data without labelled outputs.
K-Means Clustering	Students should understand the idea of clustering data into K groups, centroids, assigning points to clusters, and updating centroids. They should understand the effect of choosing different values of K.
Principal Component Analysis (PCA)	Students should understand PCA as a dimensionality-reduction technique. They should understand the idea of finding directions of maximum variance and representing high-dimensional data using fewer dimensions. They should understand why PCA can be useful for visualization and reducing dimensionality.
Other clustering methods	Recognition only. Students should recognise t-SNE, UMAP, DBSCAN and hierarchical clustering as names for dimensionality reduction or clustering. No mechanism, no parameters, no comparison questions.

Topic #3: Data Science and Model Evaluation

Training, Validation and Test Sets	Students should understand why datasets are divided into training, validation, and test sets. They should understand what information should and should not be used during model training and evaluation, and why evaluating on training data alone can be misleading.
Accuracy, Recall, Precision, F1-Score	Students should understand what these metrics are and how they are calculated.
Confusion Matrix, ROC Curves and AUC	Students should be able to interpret and understand these metrics.
Underfitting and Overfitting	Students should understand both concepts, their causes, and their effects on training and unseen data. They should be able to identify underfitting and overfitting from simple examples or performance curves.
Bias and Variance	Students should have a basic conceptual understanding of high bias and high variance and their relationship to underfitting and overfitting.

Topic #4: Neural Networks and Deep Learning

Neural Network Basics	Students should understand the basic structure of an artificial neuron, including inputs, weights, bias, activation, and output. They should understand how neurons are combined into layers to form neural networks.
Perceptron	Students should understand the perceptron as a simple model that combines weighted inputs and produces an output using a decision rule/activation function. They should understand its limitations, particularly for non-linearly separable problems.
Forward Propagation	Students should understand how input data passes through the layers of a neural network to produce an output. They should understand weights, biases, activations, and layer structure conceptually.
Gradient Descent	Students should understand the basic purpose of gradient descent: changing model parameters to reduce the loss function. They should understand the concepts of learning rate, gradient, and direction of parameter updates.
Backpropagation	Students should understand why backpropagation is needed and how the error/loss is propagated backwards through a neural network to calculate gradients used for updating weights. Detailed calculus is not required for Round 1.
Activation Functions	Students should understand the purpose of activation functions and know the basic properties and shapes of ReLU, Sigmoid, and Tanh. They should understand why nonlinear activation functions are important.
Loss Functions	Students should understand that a loss function measures prediction error and guides model training. They should know the basic purpose and characteristics of MSE, MAE, and Cross-Entropy Loss, and when each is generally used.
Multi-Layer Perceptrons (MLP)	Students should understand the structure of an MLP and how multiple fully connected layers allow a neural network to model more complex

	relationships.
Embeddings	Students should understand the basic idea of converting text, images, audio, or other data into numerical vector representations called embeddings. They should understand that embeddings can represent similarity or useful semantic information.
Pooling	Students should understand the purpose of pooling and know the difference between max pooling and average pooling. They should understand how pooling affects the size and information content of representations.
Stochastic Gradient Descent (SGD), Mini-Batch Gradient Descent	Students should understand the difference between batch, stochastic, and mini-batch gradient descent at a conceptual level.
Momentum Methods	Students should understand the basic purpose of momentum in optimization and know that Adam and AdamW are commonly used adaptive optimization methods. Detailed implementation is not required.
Learning Rate and Convergence	Students should understand the role of the learning rate and the consequences of choosing it too high or too low. They should understand the basic idea of convergence and why training may fail to converge.
Regularization	Students should understand how dropout, early stopping, and weight decay can reduce overfitting. They should know the basic idea behind each method.
Weight Initialization	Students should understand why neural-network weights should not simply be initialized identically and why appropriate initialization helps training.
Batch Normalization	Students should have a basic conceptual understanding of batch normalization and why it can help neural networks train more effectively.

Topic #5: Computer Vision

Convolutional Layers	Students should understand the basic idea of convolution, kernels/filters, feature maps, stride, and padding. They should understand why convolutions are useful for image data and how early and deeper layers can learn different visual features.

Image Classification	Students should understand the task of assigning an image to one or more categories. They should understand the basic pipeline from image → model → class prediction.
Image Augmentation	Students should understand why images are augmented during training and know common transformations such as flipping, cropping, rotation, scaling, and adding noise.

Important information

Resources

- Numpy: numpy tutorial - introduction | numpy array vs python list
  This is a course by codebasics. 6 episodes for topic 1.

- Pandas: Python Pandas Tutorial 1. What is Pandas python? Introduction and Insta…
  Again a course by codebasics. 22 episodes for topic 1

- Machine Learning Tutorial Python Series by codebasics. For Topic 2 and 3
- Deep Learning Tutorial (Codebasics) or Deep Learning Full Course (84 Videos)(campus X). for topic 4 and 5. Be aware that you need to learn about the theory part only for the first round.

Note: Only theory part is sufficient for crossing the cutoff in round 1. Studying additional topics is always motivated and may even boost your first round score.