Indian Institute of Science Education and Research Berhampur

Proposal of a New Course

Department of Computer Science

<table>
<tr><td>Course Title</td><td colspan="5">Introduction to Programming with Python</td></tr>
<tr><td>Course Level (tick (√) in the relevant option)</td><td colspan="3">100 √ 200 ☐ 300 ☐ 400 ☐ 500 ☐ 600 ☐ 700 ☐</td><td colspan="2">Semester: Odd ☐ Even √
Either semester ☐</td></tr>
<tr><td>Course applicable for</td><td colspan="3">BSMS √ Int. PhD. ☐ Ph.D. ☐</td><td colspan="2"></td></tr>
<tr><td>Tag of the course (tick (√) in the relevant option(s))</td><td colspan="5">Institute Core √ Professional (Department) Core ☐ Department Elective ☐ Open Elective ☐ Minor ☐</td></tr>
<tr><td>Type of the course</td><td colspan="5">Theory √ Lab √ Project ☐ Non-Credit ☐</td></tr>
<tr><td rowspan="2">Credit Structure</td><td>Lectures (L)</td><td>Tutorials (T)</td><td>Practicals (P)</td><td>Total Credits</td><td></td></tr>
<tr><td>2</td><td>0</td><td>1</td><td>3</td><td></td></tr>
<tr><td>Course Code:</td><td colspan="5">CS102</td></tr>
<tr><td>Names of faculties who can teach the course</td><td colspan="5">All faculty members in Computer Science Department</td></tr>
<tr><td colspan="6">Will the course require participation of visiting faculties: Yes / No√</td></tr>
<tr><td>Prerequisites, if any Course code(s)</td><td colspan="5">None</td></tr>
<tr><td colspan="6">Learning Objective(s): The objective of this course is to introduce students to the fundamental concepts of programming using the Python language. The course aims to develop a strong foundation in computational thinking, algorithmic logic, and code organization through practical, hands-on experience. It emphasizes problem-solving, code readability, and real-world applications, enabling students to write modular, efficient, and error-free Python programs.</td></tr>
<tr><td colspan="6">Expected Learning Outcome(s):
By the end of this course, students will be able to:
- Understand programming fundamentals and core concepts, including data types, control structures, functions, and object-oriented programming paradigms.
- Write and execute Python programs
- Apply control flow constructs
- Use and manipulate Python data structures
- Design reusable functions and understand variable scope
- Perform file operations with proper exception handling mechanisms in-place
- Follow best practices of Python and solve real-world problems programmatically</td></tr>
<tr><td colspan="6">Contents of the course:
Introduction to Programming: Types of Programming Languages (low-level and high-level), (compiled vs inter and high-level), compiled vs interpreted languages and their examples, programming paradigms: procedural, Object-Oriented Programming (OOP), functional programming (brief overview), Event-driven Programming
Introduction to Python Programming: Python interpreters (CPython, IDLE, Jupyter Notebook), Python environment setup (installation and configuration), .py files, single and multi-line comments, use of input() and print() functions, syntax, variables, “Hello World” Program. Intro to virtual environments (venv).
Operators: Different types of Operators: Arithmetic, bitwise, logical, assignment, comparison, identity & membership operators (is, is not, in, not in), etc. Precedence of operator.</td></tr>
</table>

Operations: Operation on String (concatenation, repetition, slicing) , list (indexing, appending, slicing, nested lists), dictionary (key access, insertion, deletion), and set (union, intersection, difference)

Data Types: Built-in data types: numeric, boolean, string, list, set, dictionary, tuple, None, etc.
Data type conversion (implicit and explicit conversion), type checking using type() and isinstance().

Control Structure: if-else, nested-if, while, for loop, break, continue statements.

Function: Modular Programming, Reusability, def keyword, naming conventions, function calls and execution flow, Arguments (positional, keyword, defaults, variable-length (*args, **kwargs), command line), variable scope and lifetime, global keyword, recursion with examples (e.g., factorial, Fibonacci), docstrings using triple quotes, lambda (anonymous) functions.

Class and Objects: Procedural and object-oriented programming, class naming conventions, class keyword classes docstring, Class Instantiation-Object, use of self-parameter, defining attributes in __init__() method, accessing and modifying attributes, child class, method overriding, use of super(), checking object type and class relationships using isinstance(), use of regular expression.

File handling: Why file?, file types and extensions (text and binary files), file operations (create, read, write, append, delete), file opening modes, open() and close() functions, content reading from file, content writing methods, use of with keyword, navigation within content.

Error and Exception Handling: Error types: syntax and runtime, common runtime errors (e.g. ZeroDivisionError, TypeError, NameError, ValueError, FileNotFoundError, PermissionError etc.), error vs exception handling, try and except blocks, catching specific exceptions, multiple except blocks, use of raise to throw an exception, else and finally clauses, and custom exceptions.

Python best practices: Naming conventions (variables, functions, classes), indentation and spacing, Line length and breaking long lines, keeping code modular (functions, classes, files), using of __init__.py) for modular organization, use of built-in functions (enumerate, zip, any, all, etc., inline vs block comments (documentation).

Lecture Outline: (Not more than 6 lectures per module)

Module No	Topic	No. of Hours
1	**Introduction to Programming and Python Basics:** Programming types and paradigms, OOPs, Python introduction, and "Hello World !!!" program. Setting up virtual environments (venv)	4
2	**Data Types, Operations, and Operators:** Data types, type checking and conversion, operator types, String operation, lists and sets operations, dictionary.	6
3	**Control Structures and Functions:** Control structures (loops and loop control), Function, def keyword, arguments, variable scope and lifetime, return statement, recursion, lambda functions.	6
4	**Object-Oriented Programming (OOP) in Python:** Procedural vs Object-Oriented Programming, class keyword, instance and class attributes, use of super() and isinstance().	6
5	**File Handling and Exception Management:** File types and extensions, file operations and modes, file navigation, exception handling, data handling using Pandas, Python best practices.	6
	TOTAL:	**28**

Brief Description of Tutorial Activities:

Module No	Topic	No. of Hours

Brief Description of Laboratory Activities:

Experiment No:	Topic	No. of Hours
1	**Python Basics and Environment Setup (syntax, variables, I/O)**: Setup: Install Python, IDE (VS Code or PyCharm), working with virtual environment, pip, Write and run “Hello, World!” program; Programming Basic:	5

	Variables & types, Expressions, String ops; I/O: Read input, type conversion, basic validation, Read/write a text file	
2	**Operators, Data Types, and Operations**: Sequences, List, Tuples, Set, dictionary	7
3	**Control Structures (Conditionals, loops) and Functions:** Branching and Ternary expression with example, For/while loop examples, Iterate over dictionary (items, keys, values), enumerate, functions, command line argument, variable arguments (*args, **kwargs), use of pytest (optional)	9
4	**OOP: Classes, objects, inheritance, file handling:** Class design, objects, Inheritance & polymorphism, Subclasses, File handling, handling CSV, json, yaml, txt files, Context managers & pathlib, Use of open(...)	12
5	**Error Handling (Exceptions, modular code), pandas and Best Practices:** Exceptions & logging, Creating custom exception, use of try/except/else/finally, logging to file and console; logging levels demo, Modularization and refactoring code into src/ with __init__.py, utils.py, pyproject.toml/requirements.txt, pandas intro (read_csv, inspect head/info/describe, select/filter, GroupBy	9
	TOTAL:	**42**

Suggested Reading: (Please use the format: Author(s) (Surname first), Title, Publisher, Year of publication.

- Sundarrajan M., Mani Deepak Choudhry, Jeevanandham S, Akshya Jothi, *Python Programming: Beginners Guide*, Notion Press, 2024
- Deepali Srivastava, *Ultimate Python Programming*, BPB Publications, 2024
- Paul Barry, *Head First Python*, 3rd edition, O'Reilly Media, 2023
- Mark Lutz, *Learning Python*, 6th Edition, O'Reilly Media, 2025
- Online resource: https://docs.python.org/3/

Modalities for Students Performance Assessment (with details of weightage for different components of performance assessment within the broad guidelines given in BS-MS/PhD manual):

The students' progress will be assessed through the following:

- **Lab work: 50%**
- **Mid-semester exam: 20%**
- **End-semester exam: 30%**

Additional Guidelines:

- Encourage the use of online learning
- Encourage the use of latest stable Python version

Name and Signature of course proposer(s):

Dr. Chinmaya Kumar Dehury

Similarity with the existing courses, if any: Nil

(Similarity content is declared as per the number of lecture hours on similar topics)

S. No.	Name of the Course	Course Code	Similarity Content	Approx. % of Content
1.	None	None	None	NA

Justification of new course proposal if cumulative similarity content is >30%:

Recommendation of External experts/Senate members/Subject expert(s)

Description (Add separate sheet, if required) *Please find attached sheet*	Description (Add separate sheet, if required) *Please find attached sheet*

The above proposal followed by the recommendations of the External experts/Senate members/Subject expert(s) appointed by Director is found to be acceptable by the DUGC/DPGC in its meeting held on_____. The committee recommended the introduction of new course for consideration of CoAA/DoAA and Director/Chairperson Senate.

Convener, DUGC	Convener, DPGC	FIC/HoD

For Use of the Academic Section

Recommended/Not Recommended to Senate Chairperson	Coordinator/Dean, Academic Affairs
Recommended / Not Recommended for the approval/ratification by the Senate Director/Chairperson Senate	

Indian Institute of Science Education and Research Berhampur

Proposal of a New Course

Department of Computer Science

Course Title	Introduction to Data Analytics: Concepts and Applications				
Course Level (tick (√) in the relevant option)	100 ☐ 200 √ 300 ☐ 400 ☐ 500 ☐ 600 ☐ 700 ☐			Semester: Odd √ Even ☐ Either semester ☐	
Course applicable for	BSMS √ Int. PhD. ☐ Ph.D. ☐				
Tag of the course (tick (√) in the relevant option(s))	Institute Core √ Professional (Department) Core ☐ Department Elective ☐ Open Elective ☐ Minor ☐				
Type of the course	Theory √ Lab √ Project ☐ Non-Credit ☐				
Credit Structure	Lectures (L)	Tutorials (T)	Practicals (P)	Total Credits	
	1	0	1	2	
Course Code:	CS201				
Names of faculties who can teach the course	Dr. Anabik Pal, Dr. Chinmaya Kumar Dehury				
Will the course require participation of visiting faculties: ~~Yes~~ / No					
Prerequisites, if any Course code(s)	CS102 - Introduction to Programming with Python				

Learning Objective(s):

This is an introductory course on beginner-level understanding of data analytics theory and practical application. The course will contain the mathematical foundation of data analytics and its implementation with Python programming. This course will provide students with some exposure to data analytics and build the necessary foundation required for data-driven scientific discovery.

Expected Learning Outcome(s):

On successful completion of this course, a student will be able to:

a) Understand different types, formats, and data structures for effective organization and representation.
b) Apply descriptive statistics and probability concepts to summarize and interpret data characteristics.
c) Clean and visualize data using standard techniques to handle missing values, outliers, and patterns.
d) Build basic machine learning models for supervised and unsupervised learning tasks.
e) Implement data analytics algorithms with Python programming.

Contents of the course:

Theory:

1. **Understanding and Organizing Data:**
   a) Types of data: Categorical, Nominal, Ordinal, Discrete, Continuous
   b) Format of data: Structured, Semi-structured, vs unstructured data
   c) Data Structures and Representation: Array, list, Dictionary, data frame, XML/JSON
2. **Statistical Thinking and Probability Foundations:**
   a. Descriptive Statistics: Central Tendency (Mean, Median, Mode), Dispersion (Range, Variance, Standard Deviation, Confidence Intervals)
   b. Probability Concepts: Random variables and basic probability, Discrete distributions (Bernoulli, Binomial), Continuous distribution (Normal distribution)

c. Statistical Inference: intuitive understanding of Hypothesis testing (t-test, p-value, and ANOVA)

3. **Data Cleaning and Visualization:**
   a. Missing Value Handling: Imputation, Deletion (dealing with incomplete data)
   b. Outlier Detection and Removal: Identifying and addressing extreme values
   c. Data Visualizations: Histograms, Boxplots, Scatterplots, Bar charts, Line charts, Pie charts
4. **Basics of Machine Learning:**
   a. Supervised Learning: Understanding and application of Regression, Classification
   b. Unsupervised Learning: Understanding and application of Clustering, Dimensionality Reduction

Practical:

1. **Python pandas, numpy packages:**
   a. Read, write data stored in .xlsx, .csv, .xml, .json file
   b. Conversion of data and transform into different form
   c. Manipulation of data for finding descriptive statistics, hypothesis testing and missing value imputation
2. **Python Matplotlib, Seaborn, Scipy packages:**
   a. Plotting data to visualize data distribution
   b. Creating and customizing histograms, bar charts, line plots, box plots, and scatter plots
3. **Python scikit-learn package:**
   a. Performing Supervised learning (KNN, Decision Trees, Random Forests, SVM) and performance evaluation
   b. Performing Clustering (K-Means, Gaussian mixture model)

Lecture Outline: (Not more than 6 lectures per module)

Module No	Topic	No. of Hours
1	Types of data: (Categorical, Nominal, Ordinal, Discrete, Continuous), Format of data: Structured, Semi-structured, vs unstructured data	1
2	Data Structures and Representation: Array, list, Dictionary, data frame, XML/JSON	2
3	Data Visualisations: Histograms, Boxplots, Scatterplots, Bar charts, Line charts, Pie charts	1
4	Descriptive Statistics: Central Tendency (Mean, Median, Mode), Dispersion (Range, Variance, Standard Deviation, Confidence Intervals)	1
5	Probability Concepts: Random variables and basic probability, Discrete distributions (Bernoulli, Binomial), Continuous distribution (Normal distribution)	2
6	Statistical Inference: intuitive understanding of Hypothesis testing (t-test, p-value, and ANOVA)	3
7	Machine Learning: Supervised learning, Unsupervised learning, Missing Value Handling (Imputation, Deletion), Outlier Detection and Removal	4
	Total Lecture Hours	14

Brief Description of Tutorial Activities:

Module No	Topic	No. of Hours

Brief Description of Laboratory Activities:

Experiment No:	Topic	No. of Hours
1.	Python revisited for list, dictionary, and array manipulation	3
2.	Usage of Python libraries: Logical & conditional functions, Lookup functions, Sorting, Filtering, Formatting, Summarization and Aggregation	3
3.	Python package (e.g. Pandas) for reading, writing data stored in .xlsx, .csv, .xml, .json files	3
4.	Python package (e.g. Pandas) for data manipulation	3
5.	Data visualization and plotting using Python packages (e.g. Matplotlib, Seaborn)	3
6.	Python-Numpy, Scipy or other packages for advanced data manipulation for finding descriptive statistics, hypothesis testing and missing value imputation, etc.	3
7.	Python Scikit-learn package for classification, clustering	12
	Total Lecture Hours	42

Suggested Reading: (Please use the format: Author(s) (Surname first), Title, Publisher, Year of publication.

1. Doing Data Science by Cathy O'Neil, Rachel Schutt Released October 2013 Publisher(s): O'Reilly Media, Inc. ISBN: 9781449358655
2. Data Science from Scratch, 2nd Edition by Joel Grus Released May 2019 Publisher(s): O'Reilly Media, Inc. ISBN: 9781492041139
3. From Concepts to Code Introduction to Data Science by Adam P. Tashman *CRC Press © 2024* ISBN: 978-1-032-51795-7
4. An Introduction to Statistical Learning with Applications in Python by Gareth James, Daniela Witten, Trevor Hastie, Rob Tibshirani, Jonathan Taylor
5. Official Documentation of Python packages (pandas, numpy, Matplotlib, Seaborn, Scipy, scikit-learn)

Modalities for Students Performance Assessment (with details of weightage for different components of performance assessment within the broad guidelines given in BS-MS/PhD manual):

Theory (Midsem 30, EndSem 30): 60,
Practical: 40

Additional Guidelines:

Name and Signature of course proposer(s):

Similarity with the existing courses, if any: N/A

(Similarity content is declared as per the number of lecture hours on similar topics)

S. No.	Name of the Course	Course Code	Similarity Content	Approx. % of Content
1.				
2				

Justification of new course proposal if cumulative similarity content is >30%:

Recommendation of External experts/Senate members/Subject expert(s)

Description (Add separate sheet, if required)	Description (Add separate sheet, if required)

The above proposal followed by the recommendations of the External experts/Senate members/Subject expert(s) appointed by Director is found to be acceptable by the DUGC/DPGC in its meeting held on_____. The committee recommended the introduction of new course for consideration of CoAA/DoAA and Director/Chairperson Senate.

Convener, DUGC Convener, DPGC	FIC/HoD

For Use of the Academic Section

Recommended/Not Recommended to Senate Chairperson	Coordinator/Dean, Academic Affairs
Recommended / Not Recommended for the approval/ratification by the Senate Director/Chairperson Senate	