Informatics Practices(New)

CLASS XII

Code No. 065

2019-20

1. Prerequisites

Informatics Practices – Class XI

2. Learning Outcomes

1. Understand aggregation operations, descriptive statistics, and re-indexing columns in a Data Frame.
2. Apply functions row-wise and element-wise on a Data Frame.
3. Understand basic software engineering: models, activities, business use-case diagrams, and version control systems.
4. Connect a Python program with a SQL database, and learn aggregation functions in SQL.
5. Have a clear understanding of cyber ethics and cybercrime. Understand the value of technology in societies, gender and disability issues, and the technology behind biometric ids.

3. Distribution of Marks

Unit No.	Unit Name	Marks
1.	Data Handling - 2	30
2.	Basic Software Engineering	15
3.	Data Management - 2	15
4.	Society, Law and Ethics - 2	10
5.	Practicals	30
	Total	100

4.1. Unit 1: Data Handling (DH-2) (80 Theory + 70 Practical)

4.1.1. Python Pandas

- Advanced operations on Data Frames: pivoting, sorting, and aggregation
- Descriptive statistics: min, max, mode, mean, count, sum, median, quartile, var
- Create a histogram, and quantiles.
- Function application: pipe, apply, aggregation (group by), transform, and apply map.
- Reindexing, and altering labels.

4.1.2. Numpy

- 1D array, 2D array
- Arrays: slices, joins, and subsets
- Arithmetic operations on 2D arrays
- Covariance, correlation and linear regression

4.1.3. Plotting with Pyplot

- Plot bar graphs, histograms, frequency polygons, box plots, and scatter plots.

4.2 Unit 2: Basic Software Engineering (BSE) (25 Theory + 10 Practical)

- Introduction to software engineering

- Software Processes: waterfall model, evolutionary model, and component based model
- Delivery models: incremental delivery, spiral delivery
- Process activities: specification, design/implementation, validation, evolution
- Agile methods: pair programming, and Scrum
- Business use-case diagrams
- Practical aspects: Version control system (GIT), and do case studies of software systems and build use-case diagrams

4.3. Unit 3: Data Management (DM-2) (20 Theory + 20 Practical)

- Write a minimal Django based web application that parses a GET and POST request, and writes the fields to a file – flat file and CSV file.
- Interface Python with an SQL database
- SQL commands: aggregation functions, having, group by, order by.

4.4. Unit 4: Society, Law and Ethics (SLE-2) (15 Theory)

- Intellectual property rights, plagiarism, digital rights management, and licensing (Creative Commons, GPL and Apache), open source, open data, privacy.
- Privacy laws, fraud; cybercrime- phishing, illegal downloads, child pornography, scams; cyber forensics, IT Act, 2000.
- Technology and society: understanding of societal issues and cultural changes induced by technology.
- E-waste management: proper disposal of used electronic gadgets.
- Identity theft, unique ids, and biometrics.
- Gender and disability issues while teaching and using computers.
- Role of new media in society: online campaigns, crowdsourcing, smart mobs
- Issues with the internet: internet as an echo chamber, net neutrality, internet addiction
- Case studies - Arab Spring, WikiLeaks, Bit coin

5. Practical

S.No.	Unit Name	Marks
1.	Lab Test (10 marks)	
	Python programs for data handling (60% logic + 20% documentation + 20% code quality)	7
	Small Python program that sends a SQL query to a database and displays the result. A stub program can be provided.	3
2.	Report File + viva(9 marks)	
	Report file: Minimum 21 Python programs. Out of this at least 4 programs should send SQL commands to a database, and retrieve the result; at least 1 program should implement the web server to write user data to a CSV file.	7
	Viva voce based on the report file	2
	Project + viva (11 marks)	
3.	Project (that uses most of the concepts that have been learnt)	8
	Project viva voce	3

5.1. Data Management: SQL+web-server

- Find the min, max, sum, and average of the marks in a student marks table.
- Find the total number of customers from each country in the table (customer ID, customer Name, country) using group by.
- Write a SQL query to order the (student ID, marks) table in descending order of the marks.
- Integrate SQL with Python by importing MYSQL dB
- Write a Django based web server to parse a user request (POST), and write it to a CSV file.

5.2. Data handling using Python libraries

- Use map functions to convert all negative numbers in a Data Frame to the mean of all the numbers.
- Consider a Data Frame, where each row contains the item category, item name, and expenditure.
    - Group the rows by the category, and print the total expenditure per category.
- Given a Series, print all the elements that are above the 75^{th} percentile.
- Given a day's worth of stock market data, aggregate it. Print the highest, lowest, and closing prices of each stock.
- Given sample data, plot a linear regression line.
- Take data from government web sites, aggregate and summarize it. Then plot it using different plotting functions of the PyPlot library.

5.3. Basic Software Engineering

- Business use-case diagrams for an airline ticket booking system, train reservation system, stock exchange
- Collaboratively write a program and manage the code with a version control system (GIT)

6. Project

The aim of the class project is to create something that is tangible and useful. This should be done in groups of 2 to 3 students, and should be started by students at least 6 months before the submission deadline. The aim here is to find a real world problem that is worthwhile to solve. Students are encouraged to visit local businesses and ask them about the problems that they are facing. For example, if a business is finding it hard to create invoices for filing GST claims, then students can do a project that takes the raw data (list of transactions), groups the transactions by category, accounts for the GST tax rates, and creates invoices in the appropriate format. Students can be extremely creative here. They can use a wide variety of Python libraries to create user friendly applications such as games, software for their school, software for their disabled fellow students, and mobile applications, Of course to do some of this projects, some additional learning is required; this should be encouraged. Students should know how to teach themselves.

If three people work on a project for 6 months, at least 500 lines of code is expected. The committee has also been made aware about the degree of plagiarism in such projects. Teachers should take a very strict look at this situation, and take very strict disciplinary action against students who are cheating on lab assignments, or projects, or using pirated software to do the same. Everything that is proposed can be achieved using absolutely free, and legitimate open source software.