UGANDA CHRISTIAN UNIVERSITY

A Centre of Excellence in the Heart of Africa

Examinations Answer Sheet Advent Semester 2025

CANDIDATE'S REGISTRATION NUMBER: S25M19/024

Non-Retake	⇨									
Retake										

ACCESS NUMBER: B35101
COLLEGE/CAMPUS: (If not Main Campus) _____
FACULTY: Engineering, Design &Technology **PROGRAM** (e.g MIT, MSDS, MSCS): MSDS1

COURSE OF EXAMINATION: Object Oriented Programming

DATE OF EXAMINATION: 12th December 2025 (As Shown on the question paper)
NB: No Answer Script shall be accepted after the Deadline
NOTE:

Uganda Christian University Integrity Covenant
As a student of the Uganda Christian University, it is my responsibility to conduct the whole of my academic career with unwavering integrity. I do this because I value integrity and because the entire scholarly enterprise is balanced on the assumption that we can trust one another.

Therefore, I pledge to act with academic integrity by;

i. Writing this examination
ii. Identifying/acknowledging the source of the ideas or words or images that I used in my work

By writing this examination I accept to be bound by this covenant, and accept the consequences if I am in breach thereof

Your Response Should

- Be typed in digital word format and submitted in Portable Document Format (.pdf)
- Use the Trebuchet MS font type; size 12, Line spacing 1.5

Submit your examination answers to the University through either e-learning platform

For Examiners Use Only

Q	I.E.	E.E.
Total Mark		

LIST QUESTIONS ANSWERED (in their Numeric order)

Project Github Link: https://github.com/TEBASIIMAJOHN/OOP_Final_Project_Exam

AUTOMATED SALES PERFORMANCE DASHBOARD

Introduction

In the business world, Sales revenue drives all most every function of a business, this requires a data driven strategy that can direct the business maneuver competition and the manage the unpredictable business external environment. Sales thus play a critical role in decision-making within fed manufacturing companies, where daily performance, customer ordering patterns, channel trends, and product movement must be monitored continuously for operational efficiency.

As businesses scale, manual tracking becomes prone to errors, delays, and inconsistencies and real time decision making becomes impossible. To address this challenge, this project develops an automated sales performance Dashboard using Python, Pandas, Plotly, and Object-Oriented Programming (OOP) principles to provide real-time sales insights. The project uses a local dataset containing daily sales submissions from different depots/channels. The system automates data cleaning, aggregation, visualization, and dashboard generation. It also integrates with Windows Task Scheduler to refresh dashboards automatically without user intervention, ensuring up-to-date performance monitoring. The project delivers a scalable, automated, and interactive reporting system, where timely data visibility is essential for operational and strategic decisions. This problem was addressed guided by the CRISP DM standard.

Problem Statement

Feed manufacturing companies rely heavily on efficient sales tracking to understand their market performance. However, the process is often hindered by it.

- Manual consolidation of sales reports from multiple selling points.
- Lack of real-time visibility in sales trends.
- Inconsistent data formats across daily submissions.

As a result, business decisions may be delayed, misinformed, based on incomplete data or even made based on personal feelings which can be inaccurate or biased. The absence of a unified automated reporting system means repeated manual work, wasted time, and inefficiencies in the sales monitoring process. This is the business pain this project seeks to alleviate using the Python and OOP principles.

Main Objective

To design and implement an automated, python-based sales performance dashboard that processes feed manufacturing sales data and provide actionable insights to decision makers.

Specific Objectives

- Implement OOP classes for data loading, cleaning, and aggregation.
- Develop a modular data processing pipeline using Pandas.
- Build interactive visualizations using Plotly.
- Automatically generate HTML dashboard files.
- Implement a refresh automation system using Windows Task Scheduler.

Company Business Model

Tunga Nutrition Uganda Limited is a feed manufacturing company that manufactures poultry and pig feeds and concentrates in Uganda. The products include broiler starters, broiler finisher, broiler grower, broiler concentrates, Layer concentrates, and pig concentrates whose demand is different across different regions in the country. The company has multiple (15) selling points in the country whose performance also differs based on region and consumer demographics. The target market of the business is farmers and feed distributers. The company generates an enormous amount of sales data from its different selling points; the data is stored online on SharePoint lists.

Dataset Description

The dataset contains daily submissions from depots with key features described below

ChannelName: Name of the outlet making the sale

ProductCategory: Category of feed (e.g., Complete Feeds, Concentrates)

Date: Date of sale

ProductName: Specific product sold (e.g., Broiler Starter, Layer Concentrate)

NetWeightKGs: Quantity sold in kilograms

NetSalesUGX: Sales value in Uganda Shillings

SalesCategory: Nature of the sales i.e., Retail or Discount sale

PaymentType: Cash or credit

CustomerName: Customer purchasing the product

This dataset offers a rich base for analyzing depot performance, product movement, customer sales behavior, and revenue trends. This being real company information, accessing it publicly required special authentication which could jeopardize the cybersecurity of the organization. The IT admin of the organization denied authenticating the live connection of Python to their SharePoint site. To address this challenge, I exported the current dataset at that time to perform offline analytics on it; to get live information into the offline analysis, I then automated the export process using Windows Task schedular to always push the updated dataset to my project folder and refresh reports. The then available dataset has 7 columns and 22,817 observations.

System Design and Architecture

Data Processing Layer (ETL)

This preprocessing involved reading, cleaning, converting data types, performing aggregations and standardizing the raw dataset to prepare it for analysis. Features engineering was also performed where date formats of month and year were added to support time series analysis. This was performed using Pandas and OOP classes of data cleaning, data aggregations and data standardization.

Visualization Layer (Dashboard)

This layer generates interactive Plotly charts and HTML files to be accessible via the browser. This made the project results accessible and shareable to different stakeholders.

Automation Layer (Refresh Engine)

In this layer, a python automation (refresh_and_export.py) was designed to automate the data extraction flow from SharePoint, this was enclosed in BATCH file for compatibility across different platforms. Windows Task Scheduler was used to refresh dashboards automatically. This was the full data pipeline adopted to automate the dashboard.

Object-Oriented Structure

The project uses OOP principles, implemented through

SalesRecord class: A lightweight object representing a single sale row.

SalesDataset class: The class encapsulates

- Data loading
- Column standardization
- Cleaning functions
- Aggregations
- KPI computations

Methods include:

- clean()
- daily_agg()
- channels_agg()
- products_agg()
- category_agg()
- compute_kpis()

PlotlyDashboard class: Responsible for:

- Bar charts
- Line charts
- Interactive filtered dashboard
- Dashboard export (write_html)

Methodology

The project followed a modular development methodology comprising the following steps

Data Importing

Using Pandas:

Raw_data = pd.read_csv("TNDailySales.csv")

The system automatically strips invisible characters (e.g., BOM, \xa0) that caused numeric conversion errors.

Data Cleaning

This cleaning pipeline performs the following actions

- Column renaming and standardization
- Date parsing (pd.to_datetime)
- Numeric conversion of NetWeightKGs & NetSalesUGX
- Removal of duplicates
- Removal of blank rows
- Adding derived fields (Month, Day, Year)

Aggregation

The system computes:

- Daily totals
- Monthly totals
- Channel-level performance
- Product performance
- Sales category performance
- KPIs: Total sales, total transactions, average price per KG

Visualizations

This is the project output that matters to stakeholders. It helps them answer key business questions such as;

1. What is the best performing outlet this month and why?
2. What is the monthly sales trend year to date?
3. Why did the business perform highly in month A and poorly in Month B?
4. What is the best-selling strategy do customers prefer (retail or discounted)?
5. What is the best-selling product this month?
6. Why did the best-selling product of last month not the best-selling this month?
7. What can be done to boost the underperforming products in best-selling points?
8. Is this business making substantial profits or not?

The project helps stakeholders answer all these questions at just a single click with dashboard visualizations. The visuals include.

- Interactive Dashboard with a Month Filter
- KPI Summary Cards
- Bar Chart: Channel vs NetWeightKGs
- Line Chart: Monthly Sales Trend
- Bar Chart: Sales Category Breakdown
- Bar Chart: Product Performance

Figure 1 depicts a part of the interactive dashboard

Automation with Task Scheduler

A standalone Python script (refresh_and_export.py) executes the entire data pipeline and exports updated dashboards to the "outputs/" directory. The .bat file wrapper enables automated execution by Task Scheduler.

Testing

To ensure the project works consistently and as intended, the tests below were ran and validated:

- Data accuracy after cleaning
- Dashboard correctness
- Auto-refresh functionality
- File export integrity
- Formatting consistency

System Implementation

Language and Tools

- Python 3
- Pandas (data manipulation)
- NumPy
- Plotly (dashboard visualizations)
- Dataclasses (OOP structure)
- Windows Task Scheduler (automation)
- Jupyter Notebook / VSCode (development)

File Outputs

The project output automatic html dashboards stored in the project output folder

- **main_dashboard.html**
- **interactive_dashboard.html**

- **cleaned dataset**

These files can be opened in any browser.

Results and Discussion

The project achieves its set objective with its dashboard delivered, the output is automated and refreshed on demand by the stakeholders.

The project helps increase the operational efficiency of Tunga Nutrition with time saved from manual repetitive data aggregations across selling points. Now decision makers refresh dashboard at just a single click.

The business is now able to monitor depot performance, track product movements, identify sales peaks and lows, and compare retail and discounted sales and monthly forecasting. All these insights help management make tactical data backed decisions.

Challenges Encountered

The best approach to realize this project was to use a cloud dataset that could make refreshes seamless, however, this was not possible with this solution due to data privacy purposes. I wanted to use a realistic dataset from a real company to put to practice my skills!! Nonetheless, I found a walk around this bottleneck and got the job done still.

At first Plotly graphs were not rendering which I later found to be caused by incorrect aggregations, this was fixed by standardizing columns before cleaning. On automating the pipeline, Task Scheduler was failing when calling Python to run the script, this was solved by wrapping the python script in a .bat file.

When working around these challenges, I learned new concepts through this headache and improved my troubleshooting skills. I can now write automation scripts!

Conclusion

This project demonstrates how Python and OOP concepts can be applied to real-world data analytics in the feed manufacturing sector. The system processes raw depot sales data, cleans it, aggregates it, generates visual dashboards, and automatically refreshes results using Task Scheduler. This project not only meets academic requirements but also provides a practical business solution capable of enhancing operational efficiency and supporting data-driven decision-making in feed manufacturing.

Recommendations

The project could be extended further to integrate machine learning models to predict customer churn, sales forecasting to give advanced value to stakeholders. I also observed that the project could only be used on Windows machines given the use of Task schedular in the pipeline. To unify the ingestion of this project across different operation systems, the project could be containerized using Docker and shared using Docker images. Here it will be accessible to any single user on the planet regardless of their operating system.