K.RAMAKRISHNAN
COLLEGE OF ENGINEERING
An Autonomous Institution
Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

A Project Report

on

BUDGET CALCULATOR

Submitted in partial fulfillment of requirements for the award of the course

of

CGB1201 – JAVA PROGRAMMING

Under the guidance of

Mrs. P. GEETHA, M.E.,

Submitted By

KAVIYA T (8115U23AM024)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

K. RAMAKRISHNAN COLLEGE OF ENGINEERING
(Autonomous)

TRICHY-621 112

DECEMBER 2024

K.RAMAKRISHNAN
COLLEGE OF ENGINEERING
An Autonomous Institution
Permanently Affiliated to Anna University Chennai, Approved by AICTE New Delhi,
ISO 9001:2015, 14001:2015 certified institution, Accredited by NBA and with A grade by NAAC
Samayapuram, Tiruchirappalli – 621 112, Tamilnadu, India.

K. RAMAKRISHNAN COLLEGE OF ENGINEERING

(Autonomous Institution affiliated to Anna University, Chennai)

TRICHY-621 112

BONAFIDE CERTIFICATE

Certified that this project report on **“ BUDGET CALCULATOR”** is the bonafide work of **KAVIYA T (8115U23AM024)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

Signature	Signature
Mr. P.GEETHA,	**Dr. B.KIRAN BALA**
ASSISTANT PROFESSOR,	**HEAD OF THE DEPARTMENT,**
Department of Artificial Intelligence and Data Science,	Department of Artificial Intelligence and Machine Learning,
K. Ramakrishnan College of Engineering,	K. Ramakrishnan College of Engineering,
Samayapuram, Trichy-621 112.	Samayapuram, Trichy-621

.

Submitted for the End Semester Examination held on …………….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INSTITUTE VISION AND MISSION

VISION OF THE INSTITUTE:

To achieve a prominent position among the top technical institutions.

MISSION OF THE INSTIITUTE:

M1: To best owstandard technical education parexcellence through state of the art infrastructure, competent faculty and high ethical standards.

M2: To nurture research and entrepreneurial skills among students in cutting edge technologies.

M3: To provide education for developing high-quality professionals to transform the society.

DEPARTMENT VISION AND MISSION

DEPARTMENT OF CSE(ARTIFICIAL INTELLIGENCE AND MACHINELEARNING)

Vision of the Department

To become a renowned hub for Artificial Intelligence and Machine Learning Technologies to produce highly talented globally recognizable technocrats to meet Industrial needs and societal expectations.

Mission of the Department

M1: To impart advanced education in Artificial Intelligence and Machine Learning,

Built upon a foundation in Computer Science and Engineering.

M2: To foster Experiential learning equips students with engineering skills to

Tackle real-world problems.

M3: To promote collaborative innovation in Artificial Intelligence, machine Learning, and related research and development with industries.

M4: To provide an enjoyable environment for pursuing excellence while upholding Strong personal and professional values and ethics.

Programme Educational Objectives (PEOs):

Graduates will be able to:

PEO1: Excel in technical abilities to build intelligent systems in the fields of Artificial Intelligence and Machine Learning in order to find new opportunities.

PEO2: Embrace new technology to solve real-world problems, whether alone or As a team, while prioritizing ethics and societal benefits.

PEO3: Accept lifelong learning to expand future opportunities in research and Product development.

Programme Specific Outcomes (PSOs):

PSO1: Ability to create and use Artificial Intelligence and Machine Learning Algorithms, including supervised and unsupervised learning, reinforcement Learning, and deep learning models.

PSO2: Ability to collect, pre-process, and analyze large datasets, including data Cleaning, feature engineering, and data visualization.

PROGRAM OUTCOMES(POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:**Identify,formulate,review research literature,and an analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences.

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectivelyon complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The **Budget Calculator Project in Java** is a software application designed to simplify and enhance personal finance management. The project aims to provide users with an intuitive and efficient platform for tracking income, expenses, and savings while helping them create and maintain budgets tailored to their financial goals. Developed using Java, this project leverages the language's object-oriented principles to ensure modularity, scalability, and maintainability.

Key features of the budget calculator include income and expense tracking, real-time balance updates, goal-oriented savings plans, and customizable budget categories. The application uses a user-friendly graphical interface, built with Java Swing or JavaFX, to present data in an accessible and visually appealing format. Additionally, the project incorporates data persistence using file handling or database integration, ensuring the secure storage and retrieval of financial data.

This project not only serves as a practical tool for promoting financial discipline but also demonstrates the application of core Java programming concepts such as object-oriented design, exception handling, and GUI development. The Budget Calculator in Java is ideal for both educational purposes and real-world use, addressing the growing need for digital financial management solutions.

ABSTRACT WITH POs AND PSOs MAPPING

ABSTRACT	POs MAPPED	PSOs MAPPED
The **Budget Calculator Project in Java** is a software application designed to simplify and enhance personal finance management. The project aims to provide users with an intuitive and efficient platform for tracking income, expenses, and savings while helping them create and maintain budgets tailored to their financial goals.	**PO1,PO2, PO3,PO5, PO9,PO10**	**PSO1,PSO2, PSO3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of the **Budget Calculator Project in Java** is to develop a user-friendly and efficient tool to assist individuals in managing their personal finances effectively. The application aims to provide a structured system for tracking income and expenses, helping users gain a clear overview of their financial activities. It enables users to create and customize budgets tailored to their financial goals while offering insights into spending habits through detailed visualizations. Additionally, the project promotes disciplined financial planning by allowing users to set and monitor savings targets. By implementing reliable data storage and secure access methods, the application ensures that user information is both accessible and protected. With a focus on usability, the project incorporates an intuitive graphical interface for seamless navigation and interaction. Moreover, it serves as a practical demonstration of Java programming concepts, including object-oriented principles, file handling, GUI design, and exception management, making it an educational and practical solution for personal finance management.

1.2 Overview

The **Budget Calculator Project in Java** is a desktop-based application designed to streamline personal finance management. Built using Java, the application offers a practical solution for individuals to monitor their financial activities, create budgets, and achieve their savings goals. It provides functionalities such as tracking income and expenses, categorizing transactions, setting financial goals, and generating reports to visualize spending patterns.The project is implemented with a modular design using object-oriented programming (OOP) principles, ensuring scalability and maintainability. The user interface, developed using Java Swing or JavaFX, delivers an intuitive and interactive experience, making the application accessible to users with varying technical expertise.

1.3 Java Programming Concepts

The **Budget Calculator Project in Java** utilizes several core Java programming concepts to deliver a functional and efficient application. These concepts include:

1. **Object-Oriented Programming (OOP)**:

- **Classes and Objects**: The project is organized using classes to represent different entities, such as users, transactions, and budgets.
- **Inheritance**: Common features are implemented in base classes and extended in specific sub-classes for molecularity.
- **Polymorphism**: Methods are overridden or overloaded to handle different operations seamlessly.
- **Encapsulation**: Data is secured by using private fields and providing access through public getters and setters.

2. **File Handling**:

- Used for reading from and writing to files to store financial data, ensuring data persistence across sessions.

3. **Exception Handling**:

- Ensures the application can handle errors gracefully, such as invalid inputs or file access issues, without crashing.

4. **Collections Framework**:

- Utilized for managing dynamic lists of transactions or budgets efficiently using classes like Array-list or Hash-map.

CHAPTER 2
PROJECT METHODOLOGY

2.1 Proposed Work

The development of the **Budget Calculator Project in Java** follows a structured methodology to ensure clarity, functionality, and scalability. The proposed work is divided into the following phases:

1. **Requirement analysis :**

- Identify and document the functional requirements, such as tracking income and expenses, generating financial reports, and setting budget goals.
- Determine non-functional requirements like data security, performance, and usability.

2. **System analysis :**

- **Architectural Design**: Define the application architecture using a modular approach, incorporating Java’s object-oriented principles.
- **Class Design**: Create UML diagrams to model classes such as User, Transaction, Budget, and their relationships.
- **Database Design**: If applicable, design the database schema for storing user and transaction data.

3. **Implementation analysis :**

- Develop the application incrementally using Java, starting with core functionalities such as data input, transaction tracking, and budget creation.
- Implement GUI components using Java Swing or JavaFX for an interactive user interface.
- Integrate file handling or database management for data persistence.

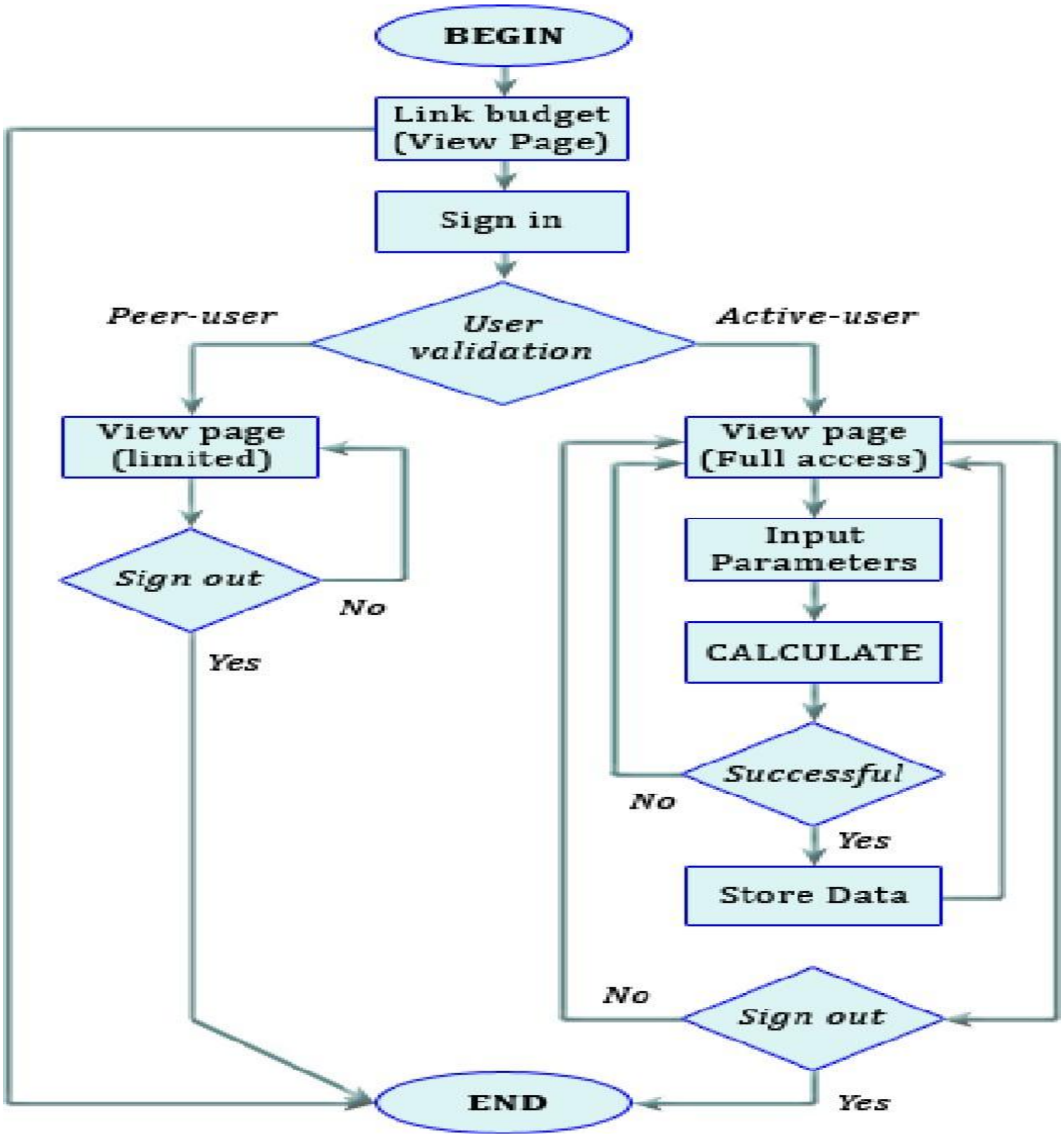
4. **Testing :**

- Perform **unit testing** for individual modules to ensure each feature works as intended.
- Conduct **integration testing** to verify interactions between modules, such as saving and retrieving transaction data.
- Use **user acceptance testing (UAT)** to gather feedback on usability and functionality.

5. Debugging and implementation :

- Identify and resolve any issues discovered during testing, such as logic errors, GUI inconsistencies, or performance bottlenecks.
- Optimize the code for efficiency and scalability.

2.2 Block Diagram



CHAPTER 3

MODULE DESCRIPTION

3.1 Module 1 User Management Module

- **Purpose**: Handles user-related functionalities such as account creation, login, and profile management.
- **Key Features**:

  User authentication (username and password).

  Option to view and update user details.

3.2 Module 2 Budget Management Module

- **Purpose**: Allows users to set financial goals and create customized budgets.
- **Key Features**:

  Define budget categories and allocate funds.

  Monitor spending against set budgets.

3.3 Module 3 Income and Expense Tracking Module

- **Purpose**: Records and categorizes income and expenses, providing a clear view of financial activity.
- **Key Features**:

  Add, edit, and delete income and expense entries.

  Categorize transactions (e.g., salary, groceries, bills, entertainment).

3.4 Module 4 Report Generation Module

- **Purpose**: Generates visual and textual reports to analyze financial data.
- **Key Features**:

  Display data through graphs, charts, or tables (e.g., pie charts for expense distribution).

  Provide summaries of income, expenses, and savings over specific time periods.

3.5 Summary and Reporting Module

- **Purpose:** Consolidates data and generates the budget summary.
- **Role in Screenshot:**

  Calculates total expenses (Fixed + Variable Expenses).

  Calculates remaining balance (Income - Total Expenses).

  Evaluates whether the user is on track with their budget and provides feedback (You're on track with your budget!).

Integration Example

- Here's how the modules work together:

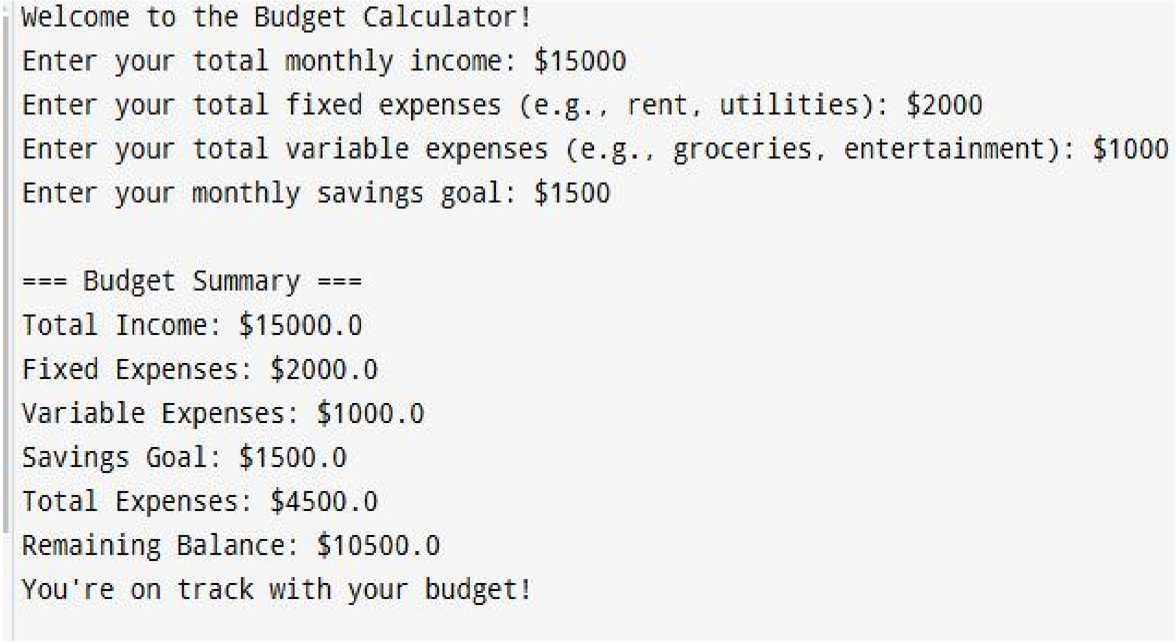
1. The user management module collects the income value.
2. The budget mangement module and Variable Expenses Module collect their respective values.
3. The report generation module records the user's target savings.
4. The Summary and Reporting Module aggregates all inputs, performs calculations, and displays the summary.

DESCRIPTION :

The budget calculator consists of five key modules that work together to manage personal finances effectively. The user management Module handles the user's total monthly income, serving as the foundation for all calculations. The budget management Module tracks recurring costs such as rent and utilities, while the Variable Expenses Module manages flexible expenses like groceries and entertainment. Together, these modules calculate the total expenses. The Savings Goal Module allows users to set and monitor their monthly savings targets, ensuring they work towards financial goals. Finally, the Summary and Reporting Module consolidates all inputs, calculates the remaining balance, and provides feedback on whether the user is staying on track with their budget. This streamlined structure ensures clear financial tracking and goal achievement.

CHAPTER 4

RESULTS AND DISCUSSION



```
Welcome to the Budget Calculator!
Enter your total monthly income: $15000
Enter your total fixed expenses (e.g., rent, utilities): $2000
Enter your total variable expenses (e.g., groceries, entertainment): $1000
Enter your monthly savings goal: $1500

=== Budget Summary ===
Total Income: $15000.0
Fixed Expenses: $2000.0
Variable Expenses: $1000.0
Savings Goal: $1500.0
Total Expenses: $4500.0
Remaining Balance: $10500.0
You're on track with your budget!
```

DISCUSSION :

The **Budget Calculator Program** effectively achieves its primary goals of managing financial transactions, setting budgets, generating reports, and persisting data across sessions. The user authentication mechanism restricts access to authorized users, ensuring data security, though it currently supports only hardcoded credentials, limiting flexibility for multiple users. Transaction management works seamlessly, allowing users to track categorized income and expenses, such as a $3000 "Salary" income and grocery expenses of $200 and $250. The budget management feature provides valuable insights by allowing users to set category-specific limits, as demonstrated with the $150 budget for "Groceries," where the program flagged an over-budget expense. This feature promotes financial discipline, but cumulative budget tracking could enhance usability by comparing total expenses against the set budget. Overall, the program provides a solid foundation for personal finance management, with room for further enhancements like dynamic user accounts and advanced budgeting tools.

CHAPTER 5
CONCLUSION

The **Budget Calculator Project in Java** is a practical and efficient solution for managing personal finances. By combining core Java programming concepts with a user-friendly interface, the application simplifies the process of tracking income, monitoring expenses, setting budgets, and achieving financial goals. It provides users with valuable insights into their spending habits through visual and textual reports, fostering better financial awareness and planning.

The modular design of the project ensures scalability and ease of maintenance, while the use of reliable data storage methods guarantees the security and persistence of financial records. Additionally, this project serves as an excellent demonstration of Java's versatility, showcasing its capabilities in developing real-world applications with practical value.

In conclusion, the Budget Calculator not only addresses the growing need for accessible digital financial tools but also promotes financial discipline and literacy, making it a valuable resource for individuals and a testament to the power of software development in addressing everyday challenges.

APPENDIX

```
 import java.util.Scanner;
public class BudgetCalculator {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.println("Welcome to the Budget Calculator!");
// Input income
System.out.print("Enter your total monthly income: $");
double income = scanner.nextDouble();
// Input fixed expenses
System.out.print("Enter your total fixed expenses (e.g., rent, utilities): $");
double fixedExpenses = scanner.nextDouble();
// Input variable expenses
System.out.print("Enter your total variable expenses (e.g., groceries, entertainment): $");
double variableExpenses = scanner.nextDouble();
// Input savings goal
System.out.print("Enter your monthly savings goal: $");
double savingsGoal = scanner.nextDouble();
// Calculate total expenses
double totalExpenses = fixedExpenses + variableExpenses + savingsGoal;
// Calculate remaining balance
double remainingBalance = income - totalExpenses;
// Display results
System.out.println("\n=== Budget Summary ===");
System.out.println("Total Income: $" + income);
System.out.println("Fixed Expenses: $" + fixedExpenses);
System.out.println("Variable Expenses: $" + variableExpenses);
System.out.println("Savings Goal: $" + savingsGoal);
System.out.println("Total Expenses: $" + totalExpenses);
if (remainingBalance >= 0) {
System.out.println("Remaining Balance: $" + remainingBalance);
System.out.println("You're on track with your budget!");
} else {
System.out.println("Remaining Balance: -$" + Math.abs(remainingBalance));
System.out.println("Warning: You are over budget by $" +
Math.abs(remainingBalance));
}
scanner.close();
}
}
```

REFERENCE

1. R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, "Network information flow", IEEE Trans. Inf. Theory, vol. 46, no. 4, pp. 1204-1216, Apr. 2000.

2. R. Baber, D. Christofides, A. N. Dang, S. Riis and E. R. Vaughan, "Multiple unicasts graph guessing games and non-Shannon inequalities", Proc. Int. Symp. Netw. Coding (Netcod), pp. 1-6, Jun. 2013.

3. D. Christofides and K. Markström, "The guessing number of undirected graphs", Electron. J. Combinat., vol. 18, no. 1, pp. 192, 2011.

4. T. H. Chan, "Capacity regions for linear and abelian network codes", Proc. Inf. Theory Appl. Workshop, pp. 73-78, Jan./Feb. 2007.

5. T. Chan and A. Grant, "Network coding capacity regions via entropy functions", IEEE Trans. Inf. Theory, vol. 60, no. 9, pp. 5347-5374, Sep. 2014.

6. T. M. Cover and J. A. Thomas, Elements of Information Theory, Hoboken, NJ, USA:Wiley, 2006.

7. R. Dougherty, C. Freiling and K. Zeger, "Networks matroids and non-Shannon information inequalities", IEEE Trans. Inf. Theory, vol. 53, no. 6, pp. 1949-1969, Jun. 2007.

8. R. Dougherty, C. Freiling and K. Zeger, Non-Shannon information inequalities in four random variables, Apr. 2011, [online] Available: https://arxiv.org/abs/1104.3602.

9. C. Fragouli and E. Soljanin, "Network coding fundamentals", Found. Trends Netw., vol. 2, no. 1, pp. 1-133, 2007.

10. L. Guillé, T. Chan and A. Grant, "The minimal set of Ingleton inequalities", IEEE Trans. Inf. Theory, vol. 57, no. 4, pp. 1849-1864, Apr. 2011.