1. Expense Tracker

Description: Create a simple expense tracker where users can log their daily expenses, view a summary of spending by category, and check if they are over budget. This program uses arrays to store expenses and categories, loops to iterate over data, and conditions to check budget limits.

Features:

- Add expenses with categories (e.g., food, transport, utilities).
- Calculate total expenses and display a breakdown by category.
- Set a monthly budget and warn the user if spending exceeds it.

Code Outline:

```
java
Copy code
import java.util.Scanner;

public class ExpenseTracker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String[] categories = {"Food", "Transport", "Utilities",
"Entertainment"};
        double[] expenses = new double[categories.length];
        double budget;

        System.out.print("Set your monthly budget: ");
        budget = scanner.nextDouble();

        for (int i = 0; i < categories.length; i++) {
            System.out.print("Enter expenses for " + categories[i] + ": ");
            expenses[i] = scanner.nextDouble();
        }

        double totalExpenses = 0;
        for (double expense : expenses) {
            totalExpenses += expense;
        }

        System.out.println("\nExpense Summary:");
        for (int i = 0; i < categories.length; i++) {
            System.out.println(categories[i] + ": " + expenses[i]);
        }
        System.out.println("Total Expenses: " + totalExpenses);

        if (totalExpenses > budget) {
            System.out.println("Warning: You have exceeded your budget!");
        } else {
            System.out.println("You are within budget.");
        }

        scanner.close();
    }
}
```

2. To-Do List Application

Description: A to-do list application that allows users to add, view, and delete tasks. Arrays store tasks, loops handle viewing and listing tasks, and conditions manage adding and deleting tasks.

Features:

- Add new tasks to the list.
- View all tasks with task numbering.
- Mark tasks as completed or delete tasks from the list.

Code Outline:

```
java
Copy code
import java.util.Scanner;

public class ToDoList {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String[] tasks = new String[100];
        int taskCount = 0;

        while (true) {
            System.out.println("\nTo-Do List:");
            System.out.println("1. Add Task");
            System.out.println("2. View Tasks");
            System.out.println("3. Delete Task");
            System.out.println("4. Exit");

            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();
            scanner.nextLine();  // Clear newline

            if (choice == 1) {
                if (taskCount < tasks.length) {
                    System.out.print("Enter task description: ");
                    tasks[taskCount] = scanner.nextLine();
                    taskCount++;
                } else {
                    System.out.println("Task list is full.");
                }
            } else if (choice == 2) {
                System.out.println("\nYour Tasks:");
                for (int i = 0; i < taskCount; i++) {
                    System.out.println((i + 1) + ". " + tasks[i]);
                }
            } else if (choice == 3) {
                System.out.print("Enter task number to delete: ");
                int taskNumber = scanner.nextInt();
                if (taskNumber > 0 && taskNumber <= taskCount) {
                    for (int i = taskNumber - 1; i < taskCount - 1; i++) {
                        tasks[i] = tasks[i + 1];
                    }
                    tasks[--taskCount] = null;
                    System.out.println("Task deleted.");
                } else {
                    System.out.println("Invalid task number.");
                }
            } else if (choice == 4) {
                break;
            } else {
                System.out.println("Invalid option.");
            }
        }

        scanner.close();
    }
}
```

3. Library Book Management System

A `switch` statement is applied to help users choose between viewing available books, borrowing, returning, and exiting the program.

```
java
Copy code
import java.util.Scanner;

public class LibraryManagementSystem {
    public static void main(String[] args) {
        String[] books = {"The Great Gatsby", "1984", "To Kill a
Mockingbird", "The Catcher in the Rye"};
        boolean[] isAvailable = {true, true, true, true};
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nLibrary Management System:");
            System.out.println("1. View Available Books");
            System.out.println("2. Borrow a Book");
            System.out.println("3. Return a Book");
            System.out.println("4. Exit");

            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.println("Available Books:");
                    for (int i = 0; i < books.length; i++) {
                        if (isAvailable[i]) {
                            System.out.println((i + 1) + ". " + books[i]);
                        }
                    }
                    break;

                case 2:
                    System.out.print("Enter book number to borrow: ");
                    int borrowBookNumber = scanner.nextInt();
                    if (borrowBookNumber > 0 && borrowBookNumber <=
books.length && isAvailable[borrowBookNumber - 1]) {
                        isAvailable[borrowBookNumber - 1] = false;
                        System.out.println("You borrowed \"" +
books[borrowBookNumber - 1] + "\".");
                    } else {
                        System.out.println("Book not available or invalid
selection.");
                    }
                    break;

                case 3:
                    System.out.print("Enter book number to return: ");
                    int returnBookNumber = scanner.nextInt();
                    if (returnBookNumber > 0 && returnBookNumber <=
books.length && !isAvailable[returnBookNumber - 1]) {
```

```
                        isAvailable[returnBookNumber - 1] = true;
                        System.out.println("You returned \"" + 
books[returnBookNumber - 1] + "\".");
                    } else {
                        System.out.println("Invalid selection or book already 
available.");
                    }
                    break;

                case 4:
                    System.out.println("Exiting Library Management System.");
                    scanner.close();
                    return;

                default:
                    System.out.println("Invalid option. Please try again.");
            }
        }
    }
}
```

4. Inventory Management for Small Shop

This code version incorporates a `switch` statement to handle inventory options more effectively, such as viewing products, selling, restocking, and exiting.

```
java
Copy code
import java.util.Scanner;

public class InventoryManagement {
    public static void main(String[] args) {
        String[] products = {"Apple", "Banana", "Milk", "Bread"};
        int[] quantities = {50, 40, 30, 20};
        double[] prices = {1.00, 0.50, 2.00, 1.50};
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nInventory Management");
            System.out.println("1. View Inventory");
            System.out.println("2. Sell Product");
            System.out.println("3. Restock Product");
            System.out.println("4. Exit");

            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.println("\nProduct List:");
                    for (int i = 0; i < products.length; i++) {
                        System.out.println(products[i] + " - $" + prices[i] +
" (Stock: " + quantities[i] + ")");
                    }
                    break;

                case 2:
                    System.out.print("Enter product number to sell: ");
                    int sellProductNumber = scanner.nextInt();
                    if (sellProductNumber > 0 && sellProductNumber <=
products.length && quantities[sellProductNumber - 1] > 0) {
                        quantities[sellProductNumber - 1]--;
                        System.out.println("Sold 1 " +
products[sellProductNumber - 1]);
                    } else {
                        System.out.println("Product out of stock or invalid
selection.");
                    }
                    break;

                case 3:
                    System.out.print("Enter product number to restock: ");
                    int restockProductNumber = scanner.nextInt();
                    System.out.print("Enter quantity to add: ");
                    int quantityToAdd = scanner.nextInt();
                    if (restockProductNumber > 0 && restockProductNumber <=
products.length && quantityToAdd > 0) {
                        quantities[restockProductNumber - 1] +=
quantityToAdd;
                        System.out.println("Added " + quantityToAdd + " to "
+ products[restockProductNumber - 1]);
                    } else {
                        System.out.println("Invalid selection or quantity.");
                    }
                    break;

                case 4:
                    System.out.println("Exiting Inventory Management.");
                    scanner.close();
                    return;

                default:
                    System.out.println("Invalid option. Please try again.");
            }
        }
    }
}
```

5. Student Grade Manager

This program lets you manage and view students' grades. You can add students, input their grades, calculate their average score, and determine if they passed based on a passing grade threshold.

```
java
Copy code
import java.util.Scanner;

public class GradeManager {
    public static void main(String[] args) {
        String[] students = new String[100];
        double[] grades = new double[100];
        int studentCount = 0;
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nStudent Grade Manager:");
            System.out.println("1. Add Student");
            System.out.println("2. Input Grade");
            System.out.println("3. View All Grades");
            System.out.println("4. Calculate Average Grade");
            System.out.println("5. Exit");

            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();
            scanner.nextLine();  // Clear newline

            switch (choice) {
                case 1:
                    System.out.print("Enter student name: ");
                    students[studentCount] = scanner.nextLine();
                    grades[studentCount] = 0;  // Initialize grade to 0
                    studentCount++;
                    System.out.println("Student added.");
                    break;

                case 2:
                    System.out.print("Enter student number to input grade: 
");
                    int studentNumber = scanner.nextInt();
                    if (studentNumber > 0 && studentNumber <= studentCount) {
                        System.out.print("Enter grade for " + 
students[studentNumber - 1] + ": ");
                        grades[studentNumber - 1] = scanner.nextDouble();
                        System.out.println("Grade updated.");
                    } else {
                        System.out.println("Invalid student number.");
                    }
                    break;

                case 3:
                    System.out.println("\nAll Student Grades:");
                    for (int i = 0; i < studentCount; i++) {
                        System.out.println(students[i] + ": " + grades[i]);
                    }
                    break;

                case 4:
                    double totalGrades = 0;
                    for (int i = 0; i < studentCount; i++) {
                        totalGrades += grades[i];
                    }
                    double averageGrade = studentCount > 0 ? totalGrades /
studentCount : 0;
                    System.out.println("Average Grade: " + averageGrade);
                    break;

                case 5:
                    System.out.println("Exiting Grade Manager.");
                    scanner.close();
                    return;

                default:
                    System.out.println("Invalid option. Please try again.");
            }
        }
    }
}
```

6. Temperature Conversion Program

This program converts temperatures between Celsius, Fahrenheit, and Kelvin. The `switch` statement allows users to choose the type of conversion they want.

```
java
Copy code
import java.util.Scanner;

public class TemperatureConverter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nTemperature Converter:");
            System.out.println("1. Celsius to Fahrenheit");
            System.out.println("2. Fahrenheit to Celsius");
            System.out.println("3. Celsius to Kelvin");
            System.out.println("4. Kelvin to Celsius");
            System.out.println("5. Exit");

            System.out.print("Choose a conversion: ");
            int choice = scanner.nextInt();

            if (choice == 5) {
                System.out.println("Exiting Temperature Converter.");
                break;
            }
            System.out.print("Enter temperature: ");
            double temperature = scanner.nextDouble();

            switch (choice) {
                case 1:
                    System.out.println("Result: " + (temperature * 9 / 5 +
32) + " °F");
                    break;

                case 2:
                    System.out.println("Result: " + ((temperature - 32) * 5 / 9) + "
°C");
                    break;

                case 3:
                    System.out.println("Result: " + (temperature + 273.15) + " K");
                    break;

                case 4:
                    System.out.println("Result: " + (temperature - 273.15) + " °C");
                    break;

                default:
                    System.out.println("Invalid option. Please try again.");
            }
        }
        scanner.close();
    }
}
```

7. Basic Voting System

This program simulates a voting system where users can vote for different candidates. The `switch` statement is used to handle the voting process and display results.

```
java
Copy code
import java.util.Scanner;

public class VotingSystem {
    public static void main(String[] args) {
        String[] candidates = {"Alice", "Bob", "Charlie"};
        int[] votes = new int[candidates.length];
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nVoting System:");
            System.out.println("1. Vote for a Candidate");
            System.out.println("2. View Results");
            System.out.println("3. Exit");

            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.println("Candidates:");
                    for (int i = 0; i < candidates.length; i++) {
                        System.out.println((i + 1) + ". " + candidates[i]);
                    }
                    System.out.print("Vote for a candidate (1-" +
candidates.length + "): ");
                    int vote = scanner.nextInt();
                    if (vote > 0 && vote <= candidates.length) {
                        votes[vote - 1]++;
                        System.out.println("Vote recorded for " +
candidates[vote - 1]);
                    } else {
                        System.out.println("Invalid candidate number.");
                    }
                    break;

                case 2:
                    System.out.println("\nVoting Results:");
                    for (int i = 0; i < candidates.length; i++) {
                        System.out.println(candidates[i] + ": " + votes[i] +
" votes");
                    }
                    break;

                case 3:
                    System.out.println("Exiting Voting System.");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid option. Please try again.");
            }
        }
    }
```