**Implement the following projects which focus on different aspects of Java programming, including:**

Object-Oriented Programming principles

Exception handling

File operations

Data structures

Input validation

User interface design

**8. To-Do List Manager**

* Create a new folder called todoapp
* Inside the todoapp folder, create three files:
* TodoApp.java
* Task.java
* TodoList.java
* Copy the code for each file as shown above. Note that I've:
* Added package declarations to all files
* Renamed Main.java to TodoApp.java for clarity
* Put all classes in the same package (todoapp)
* Compile the program from outside the todoapp folder:

// Create a new folder called 'todoapp'

// Save all these files in the todoapp folder

/**/Create a class TodoApp in the package todoapp**

**// TodoApp.java**

package todoapp;

import java.util.Scanner;

public class TodoApp {

public static void main(String[] args) {

TodoList todoList = new TodoList();

Scanner scanner = new Scanner(System.in);

boolean running = true;

while (running) {

System.out.println("\nTo-Do List Manager");

System.out.println("1. Add Task");

System.out.println("2. Remove Task");

System.out.println("3. Toggle Task Complete");

System.out.println("4. View Tasks");

System.out.println("5. Exit");

System.out.print("Choose an option: ");

try {

int choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

System.out.print("Enter task description: ");

String description = scanner.nextLine();

todoList.addTask(description);

break;

case 2:

System.out.print("Enter task ID to remove: ");

int removeId = scanner.nextInt();

todoList.removeTask(removeId);

break;

case 3:

System.out.print("Enter task ID to toggle: ");

int toggleId = scanner.nextInt();

todoList.toggleTaskComplete(toggleId);

break;

case 4:

todoList.viewTasks();

break;

case 5:

running = false;

System.out.println("Goodbye!");

break;

default:

System.out.println("Invalid option! Please choose 1-5");

}

} catch (Exception e) {

System.out.println("Invalid input! Please enter a number.");

scanner.nextLine(); // Clear the invalid input

}

}

scanner.close();

}

}

**// Create a class Task in the package todoapp**

**// Task.java**

package todoapp;

public class Task {

private String description;

private boolean completed;

private int id;

public Task(int id, String description) {

this.id = id;

this.description = description;

this.completed = false;

}

public int getId() {

return id;

}

public String getDescription() {

return description;

}

public boolean isCompleted() {

return completed;

}

public void toggleComplete() {

completed = !completed;

}

@Override

public String toString() {

return String.format("[%d] %s %s",

id,

completed ? "[✓]" : "[ ]",

description);

}

}

**// Create TodoList class in the package todoapp**

**// TodoList.java**

package todoapp;

import java.util.ArrayList;

import java.util.List;

public class TodoList {

private List<Task> tasks;

private int nextId;

public TodoList() {

tasks = new ArrayList<>();

nextId = 1;

}

public void addTask(String description) {

tasks.add(new Task(nextId++, description));

System.out.println("Task added successfully!");

}

public void removeTask(int id) {

boolean removed = tasks.removeIf(task -> task.getId() == id);

if (removed) {

System.out.println("Task removed successfully!");

} else {

System.out.println("Task not found!");

}

}

public void toggleTaskComplete(int id) {

for (Task task : tasks) {

if (task.getId() == id) {

task.toggleComplete();

System.out.println("Task status updated!");

return;

}

}

System.out.println("Task not found!");

}

public void viewTasks() {

if (tasks.isEmpty()) {

System.out.println("No tasks in the list!");

return;

}

System.out.println("\nTo-Do List:");

System.out.println("-----------");

for (Task task : tasks) {

System.out.println(task);

}

System.out.println();

}

}

o/p

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 1

Enter task description: Class and object reading

Task added successfully!

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 4

To-Do List:

-----------

[1] [ ] Class and object reading

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 3

Enter task ID to toggle: 1

Task status updated!

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 4

To-Do List:

-----------

[1] [✓] Class and object reading

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 3

Enter task ID to toggle: 1

Task status updated!

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 4

To-Do List:

-----------

[1] [ ] Class and object reading

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: a

Invalid input! Please enter a number.

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 1

Enter task description: Inheritance Reading

Task added successfully!

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 4

To-Do List:

-----------

[1] [ ] Class and object reading

[2] [ ] Inheritance Reading

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 3

Enter task ID to toggle: 2

Task status updated!

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 4

To-Do List:

-----------

[1] [ ] Class and object reading

[2] [✓] Inheritance Reading

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 2

Enter task ID to remove: 2

Task removed successfully!

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: 4

To-Do List:

-----------

[1] [ ] Class and object reading

To-Do List Manager

1. Add Task

2. Remove Task

3. Toggle Task Complete

4. View Tasks

5. Exit

Choose an option: