NAME: SHASHWAT SHAH
SAP ID: 60004220126
DIV/BATCH: C22

DISTRIBUTED COMPUTING (DC)
EXPERIMENT 05

AIM: To implement Load-Balancing using JAVA.

CODE:
LoadBalance.java

```
import java.util.Scanner;

public class LoadBalance {

   // Method to print the load distribution across servers
   static void printLoad(int server, int processes) {
      int each = processes / server; // Processes per server
      int extra = processes % server; // Extra processes to be distributed

      // Distribute extra processes first
      for (int i = 0; i < extra; i++) {
         System.out.println("Server " + (i + 1) + " has " + (each + 1) + " processes");
      }

      // Distribute remaining processes
      for (int i = extra; i < server; i++) {
         System.out.println("Server " + (i + 1) + " has " + each + " processes");
      }
   }

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

      // Input number of servers and processes
      System.out.print("Enter the number of servers: ");
      int servers = sc.nextInt();
      System.out.print("Enter the number of processes: ");
      int processes = sc.nextInt();

      // Infinite loop for the load balancing options
      while (true) {
         // Display current load distribution
         printLoad(servers, processes);

         // Menu options for modifying servers and processes
         System.out.println("\n1. Add Servers");
         System.out.println("2. Remove Servers");
         System.out.println("3. Add Processes");
         System.out.println("4. Remove Processes");
         System.out.println("5. Exit");
         System.out.print("> ");
```

```
            // Handle user input based on choice
            switch (sc.nextInt()) {
                case 1:
                    System.out.print("Enter the number of servers to add: ");
                    servers += sc.nextInt();
                    break;

                case 2:
                    System.out.print("Enter the number of servers to remove: ");
                    servers -= sc.nextInt();
                    if (servers < 1) {
                        servers = 1; // Ensure at least one server is present
                        System.out.println("At least one server is required.");
                    }
                    break;

                case 3:
                    System.out.print("Enter the number of processes to add: ");
                    processes += sc.nextInt();
                    break;

                case 4:
                    System.out.print("Enter the number of processes to remove: ");
                    processes -= sc.nextInt();
                    if (processes < 0) {
                        processes = 0; // Ensure non-negative process count
                        System.out.println("Number of processes cannot be negative.");
                    }
                    break;

                case 5:
                    System.out.println("Exiting...");
                    sc.close();
                    System.exit(0);

                default:
                    System.out.println("Invalid choice. Please select a valid option.");
            }
        }
    }
}
```

OUTPUT:

- **Compile LoadBalance.java**

- **Execute LoadBalance.java**