Assignment - SW Engineering

Your Task

Your task is to create an application that in real-time transforms the information about objects from the server and relay them at fixed time intervals to the console. The included SAAB.jar is a server that provides the objects using a TCP/IP connection. Once the application is connected, at fixed time intervals, relay the latest current status of the objects according to certain conditions.

The objects from the server have different types (1-3), where type 1 and type 2 are of the same category and type 3 belong to the second category. In the output channel (to the console), the objects should be categorized according to type, category, and current distance from a certain point in a coordinate system. The objects coordinates are updated over time thus the application shall relay the latest status on the output at fixed time intervals.

Requirements:

- Implement using C++.
- The incoming objects categorizations shall be updated to reflect type, category, proximity to the Cartesian coordinates (150, 150).
- The application shall log errors to `std::clog`.
- In the output, which occurs every 1'500 milliseconds, the objects latest type/categorization/proximity level according to the requirements in the *Application* section below.

Example information flow:

(server) ==real-time==> (application) ==fixed-interval==> (console)

Server

The software package sent to you contains a server that via TCP/IP delivers data in text format. The server is configured to communicate on port 5463.

Configuration and Start of the Server

The server configuration file looks like this:

```
SERVERPORT=5463             # Default server port
MAP=map.gif                 # Path to map
```

To start the server from the command prompt:

```
java -classpath SAAB.jar com.saabtech.server.SAABServer
```

Server Data Specification

Data delivered from the server looks like:

`ID=<LONG>;X=<INT>;Y=<INT>;TYPE=<INT>`

The table below describes the data fields in detail.

Field	Type	Example data	Description
ID	LONG	2691882127991893	ID-number on the object
X	INT	119	X-coordinate for the object
Y	INT	227	Y-coordinate for the object
TYPE	INT	2	Type-id for the object

Example 1:

```
ID=2691882127991893;X=250;Y=150;TYPE=3
```

Example 2 - Two objects with the same category:

```
ID=2691882127234543;X=199;Y=230;TYPE=1
ID=2691882127221587;X=229;Y=310;TYPE=2
```

Application

The application will connect via TCP/IP and receive data and output, at fixed intervals, in binary format.

Application Data Specification

Once the application connects to the server, data delivered from the application every 1'500 milliseconds shall start with a preamble, a count, and information about the current status of the objects in the following format:

Parameter Name	Units	Data Type
preamble	`0xFEFF`	32-bit int
count		32-bit int

`count` is the number of objects to directly follow. Each object after the preamble and count have the following format:

Parameter Name	Data Type
Id	64-bit int
X	32-bit int
Y	32-bit int
Type	32-bit int
Color	4 bytes

Where the 4-byte color sequence representation is:

- *Red*: `0x1B 0x5B 0x31 0x6D`
- *Yellow*: `0x1B 0x5B 0x33 0x6D`
- *Green*: `0x1B 0x5B 0x32 0x6D`

Objects are colored according to:

- Category 2 objects: *Yellow* unless closer than *100* from the designated coordinate then *red*.
- Type 1 objects: *Green* unless closer than *75* from the designated coordinate then *yellow* and if closer than *50* then *red*.
- Type 2 objects: *Green* unless closer than *50* from the designated coordinate then *yellow*.