Java Assignment - Detailed Explanations

1) Explain the thread lifecycle with a diagram.

In Java, a thread goes through various stages in its lifecycle:

1. **New (Created)** - When a thread object is created using the Thread class but not yet started.
2. **Runnable** - After calling the start() method, the thread is ready to run but waits for CPU allocation.
3. **Blocked/Waiting** - A thread moves to this state if it needs to wait for a resource or another thread's signal.
4. **Timed Waiting** - When sleep() or wait(time) is called, the thread enters a temporary waiting state.
5. **Running** - When the CPU scheduler picks the thread, it moves to the running state.
6. **Terminated** - Once the execution is complete, the thread moves to the terminated (dead) state.

2) What do you mean by Thread synchronization? Give an example.

Thread synchronization in Java is used to prevent multiple threads from accessing shared resources simultaneously,
which could lead to inconsistent data. Java provides `synchronized` keyword and Locks to achieve synchronization.

Example:

```
class SharedResource {
    synchronized void printMessage(String message) {
        System.out.println("[" + message);
        try { Thread.sleep(1000); } catch (InterruptedException e) {}
        System.out.println("]");
    }
}
```

3) Explain any 5 AWT components with syntax.

Java AWT (Abstract Window Toolkit) provides various components for building GUI applications.

1. **Button**: A clickable button.
   Example: Button b = new Button("Click Me");

2. **Label**: A text field that displays non-editable text.
Example: Label l = new Label("Enter Name:");

3. **TextField**: Allows users to enter text input.
Example: TextField tf = new TextField(20);

4. **Checkbox**: Represents a toggleable checkbox.
Example: Checkbox cb = new Checkbox("Accept Terms");

5. **Frame**: A window with a title bar and border.
Example: Frame f = new Frame("My Frame");

4) What do you mean by wait(), notify(), notifyAll()? Explain with examples.

These are thread communication methods in Java:

- **wait()**: Makes a thread wait until another thread calls notify().
- **notify()**: Wakes up a single waiting thread.
- **notifyAll()**: Wakes up all waiting threads.

Example:

```
class Shared {
    synchronized void process() {
        try { wait(); } catch (InterruptedException e) {}
    }

    synchronized void resumeProcess() {
        notify();
    }
}
```

5) Explain the following with diagrams:

i) Remote Method Invocation (RMI)

RMI (Remote Method Invocation) allows Java objects to communicate over a network.
It uses a stub (client-side proxy) and a skeleton (server-side implementation) to handle remote calls.

Process:
1. Define a remote interface.
2. Implement the interface.

3. Generate stub & skeleton using `rmic`.
4. Start the RMI registry and bind objects.
5. Invoke methods from a remote client.

ii) JDBC (Java Database Connectivity)

JDBC is used to connect Java applications to a database. It involves:

1. Loading the driver: `Class.forName("com.mysql.jdbc.Driver")`
2. Establishing connection: `Connection con = DriverManager.getConnection(url, user, pass);`
3. Creating a statement: `Statement stmt = con.createStatement();`
4. Executing query: `ResultSet rs = stmt.executeQuery("SELECT * FROM table");`
5. Closing the connection: `con.close();`