Tru REPL

```
public abstract class TruExpr : TruStatement {
     /// Executes a TruStatement. Overridden in subclasses of TruExpr
     public abstract TruVal Interpret(Environment env);
}

/// Represents a value or literal in the Tru language.
public abstract class TruVal : TruExpr {
     public override TruVal Interpret(Environment env) {
          return this;
     }
}

public class TruBool : TruVal {
     public bool val;
     public TruBool(bool val) { this.val = val; }
}

/// Represents a callable, ie built-in or functions.
public abstract class TruCallable : TruVal {
     public abstract TruVal call(Environment env, TruExpr[] args);
}

/// Represents a built-in function, and, or, not
public class TruBuiltIn : TruCallable {
     /// TruOperation takes a TruExpr[] so do short-circuit eval.
     public delegate TruVal TruOperation(Environment env, TruExpr[]
parameters);
     public TruOperation op;

     public override TruVal call(Environment env, TruExpr[] args) {
          return this.op(env, args);
     }
}

/// Represents a user defined function.
public class TruFunc : TruCallable {
     public string[] parameters;
     public TruExpr body;
     public Environment env; // closures.

     public override TruVal call(Environment env, TruExpr[] args) {
          if (this.parameters.Length == args.Length) {
               Environment locEnv = this.env;
               // add the args to the environment.
```

```
			for (int i = 0; i < args.Length; i++) {
				locEnv = locEnv.ExtendLocal(this.parameters[i], 
args[i].Interpret(env));
			}
			return this.body.Interpret(locEnv);
		} else {
			throw new TruRuntimeException(“Function call with 
incorrect argument count");
		}
	}
}

public class TruId : TruExpr {
	public string name;
}

/// Represents a call to a function or a built-in
public class TruCall : TruExpr {
	public TruExpr func;
	public TruExpr[] args;

	public override TruVal Interpret(Environment env) {
		TruVal funcVal = this.func.Interpret(env);

		if (funcVal is TruCallable callable) {
			return callable.call(env, this.args);
		} else {
			throw new TruRuntimeException(
				$"'{funcVal}' is not callable.");
		}
	}
}

/// Represents a lambda expression (which should eval to a TruFunc)
public class TruLambda : TruExpr {
	public string[] parameters;
	public TruExpr body;

	public override TruVal Interpret(Environment env) {
		return new TruFunc(this.parameters, this.body, env);
	}
}
```

C# Basic Chat

```
public class ClientModel
{
     private TcpClient _socket;
     public string message;
     public string messageBoard;
     public bool active;
     private string _userName;

     public ClientModel (string username)
     {
          // initializing variables
          _socket = new TcpClient("127.0.0.1", 8888);
          _userName = username;
          active = true;

          // welcome message and creating the thread to handle input
          Console.WriteLine("Welcome to the conversation. Type in a 
message when ready.");

          // assign a thread to constantly be running GetMessage
          var thread = new Thread(GetMessage);
          thread.Start();

          // initial connection
          _socket.WriteString(username);
     }
}
```