

Get {  } Java™

Intro

Created by Denis Markov

# Contents

**1.Goals**

**2.Programming**

**3.Java World**

**4.Instruments and Setup**

## Goals

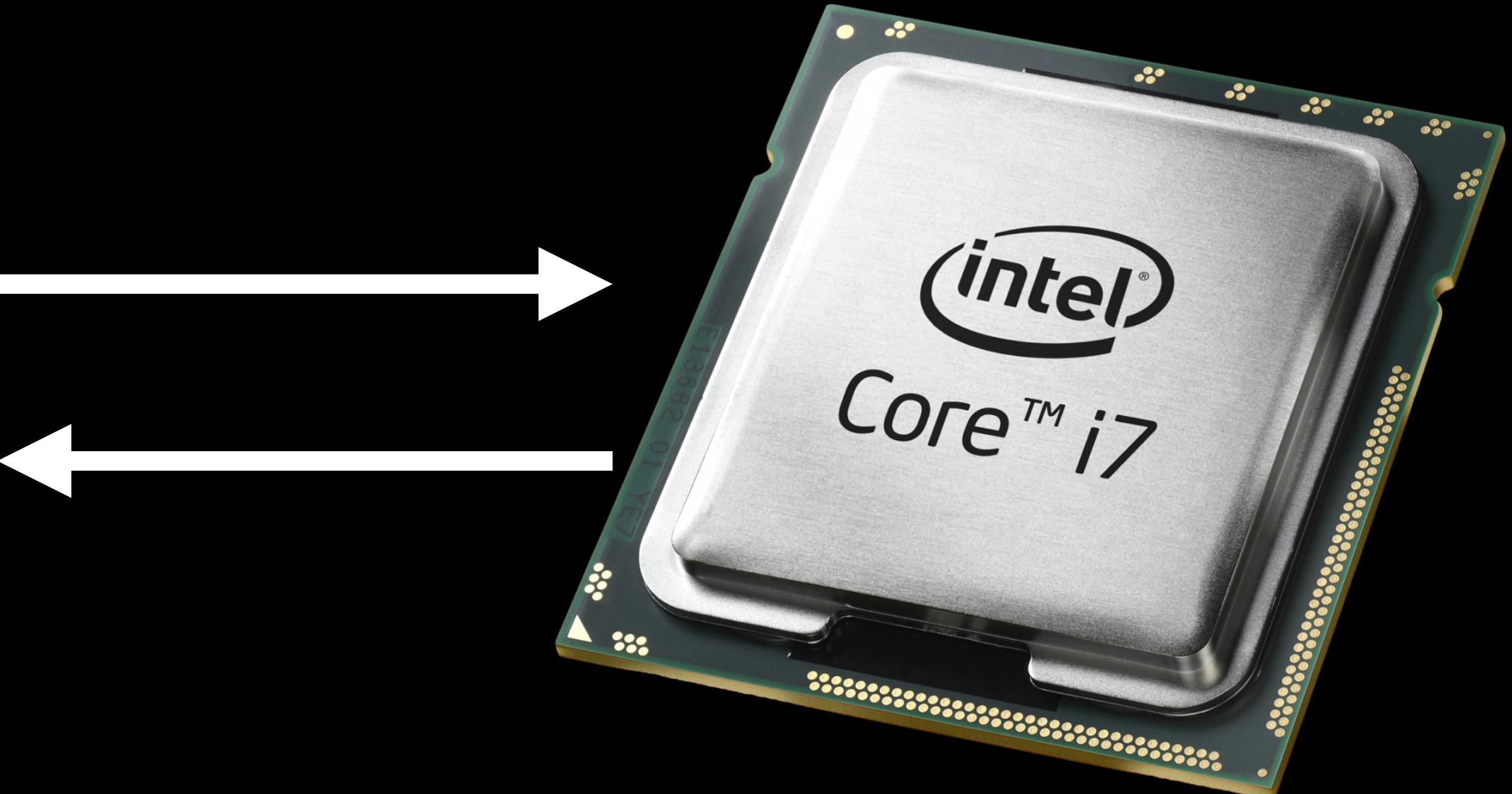
- 1. Relevant knowledge**
- 2. Completed course**
- 3. Open and Free**
- 4. Practical exercises**
- 5. Community**

# Programming

0089814E	57	PUSH EDI
0089814F	6A 00	PUSH 0
00898151	FF95 57040000	CALL DWORD PTR SS:[EBP+457]
00898157	8938	MOU DWORD PTR DS:[EAX],EDI
00898159	8D78 04	LEA EDI,[EAX+4]
0089815C	8D85 77040000	LEA EAX,[EBP+477]
00898162	57	PUSH EDI
00898163	50	PUSH EAX
00898164	E8 46000000	CALL Decompress
00898169	57	PUSH EDI
0089816A	E8 EB000000	CALL PrepareImage
0089816F	89C6	MOU ESI,EAX
00898171	31C0	XOR EAX,EAX
00898173	83EF 04	SUB EDI,4
00898176	8B0F	MOU ECX,DWORD PTR DS:[EDI]
00898178	57	PUSH EDI
00898179	F3:AA	REP STOS BYTE PTR ES:[EDI]
0089817B	5F	POP EDI
0089817C	57	PUSH EDI
0089817D	FF95 5B040000	CALL DWORD PTR SS:[EBP+45B]
00898183	8BBD 67040000	MOU EDI,DWORD PTR SS:[EBP+467]
00898189	31C0	XOR EAX,EAX
0089818B	83EF 04	SUB EDI,4
0089818E	8B0F	MOU ECX,DWORD PTR DS:[EDI]
00898190	57	PUSH EDI
00898191	F3:AA	REP STOS BYTE PTR ES:[EDI]
00898193	5F	POP EDI
00898194	57	PUSH EDI
00898195	FF95 5B040000	CALL DWORD PTR SS:[EBP+45B]
0089819B	85F6	TEST ESI,ESI
0089819D	74 07	JE SHORT 008981A6
0089819F	897424 1C	MOU DWORD PTR SS:[ESP+1C],ESI
008981A3	61	POPAD
008981A4	FFEO	JMP EAX
008981A6	61	POPAD
008981A7	B8 FFFFFFFF	MOU EAX,-1
008981AC	C2 0800	RETN 8



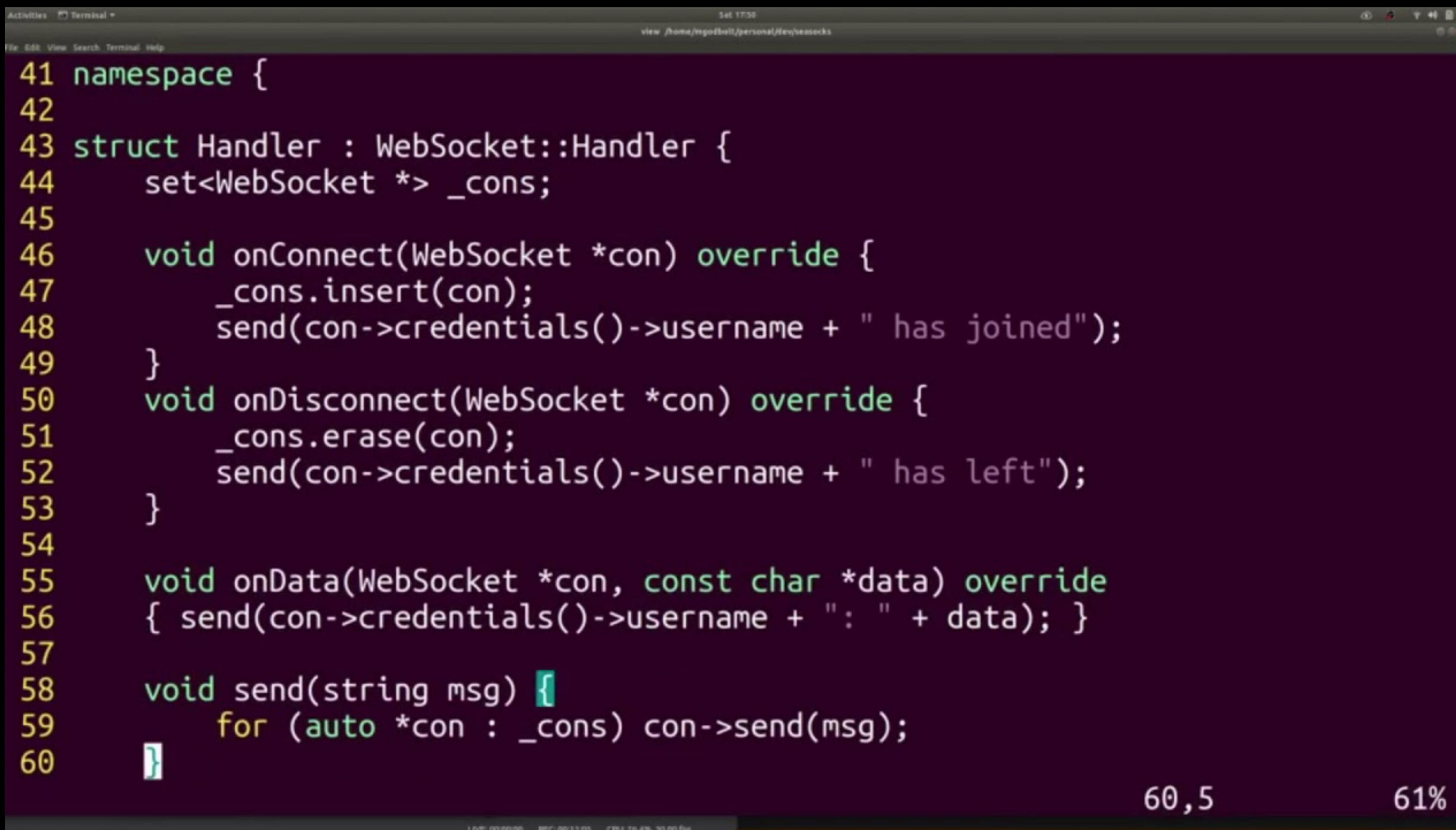
# Programming



```
{  
p000:    ; interrupt code  
        push ax  
        push ds  
        pushf  
  
        push cs  
        pop ds          ; make ds=cs  
        mov ax,wait     ; check wait time  
        dec ax          ; zero?  
        jz p010         ; yes - 1 second has elapsed  
        mov wait,ax     ; not this time  
        jmp p080        ; return
```

}

# Programming

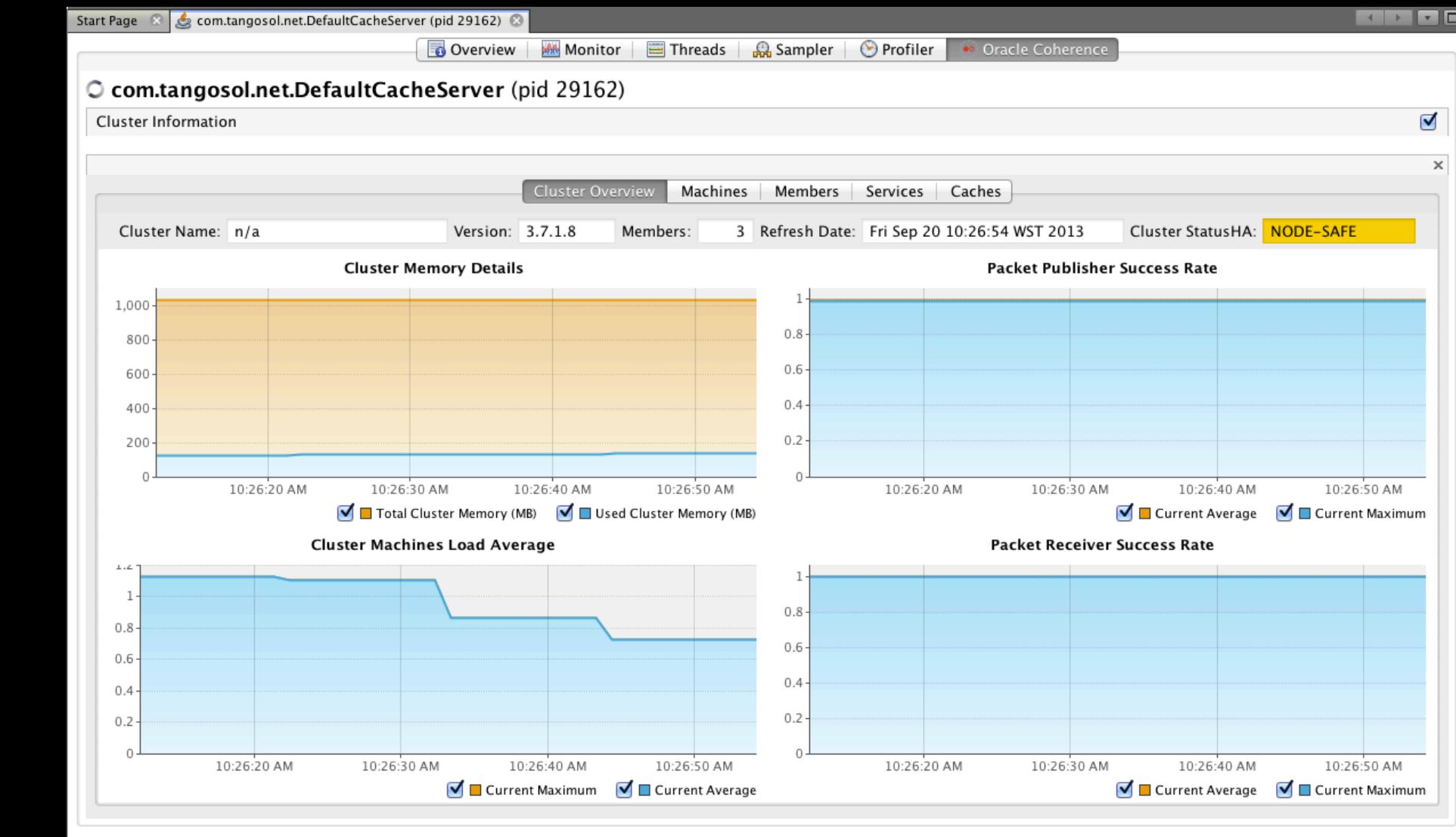


A screenshot of a terminal window titled "Terminal" showing a C++ program. The code defines a class Handler that inherits from WebSocket::Handler. It contains methods for handling connections, sending data, and disconnecting. The code uses C++11 features like move semantics and range-based loops.

```
41 namespace {
42
43     struct Handler : WebSocket::Handler {
44         set<WebSocket *> _cons;
45
46         void onConnect(WebSocket *con) override {
47             _cons.insert(con);
48             send(con->credentials()->username + " has joined");
49         }
50         void onDisconnect(WebSocket *con) override {
51             _cons.erase(con);
52             send(con->credentials()->username + " has left");
53         }
54
55         void onData(WebSocket *con, const char *data) override
56         { send(con->credentials()->username + ": " + data); }
57
58         void send(string msg) {
59             for (auto *con : _cons) con->send(msg);
60         }
61 }
```

\$ gcc main.cpp -lstdc++

# Java World



ORACLE®



- 1.Cross-Platform**
- 2.Modern Framework**
- 3.Big and live community**
- 4.Fast**
- 5.Clear architecture**
- 6.Easy to use**

# Java World

## MyDaoClass.java

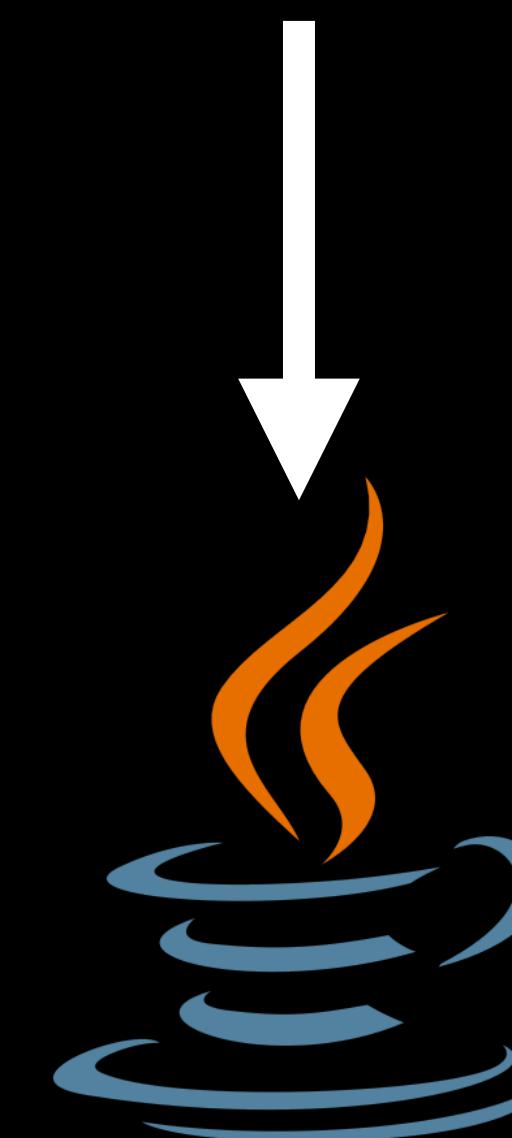
```
1 package example;
2
3 public class MyDaoClass {
4     private List<Statistics> readStatFromDatabase(String statName, long msFrom, long msTo) throws SQLException {
5         String sql = "select * from Statistics s where s.StatisticName = ? and s.StatisticTimestamp between ? and ?";
6
7         PreparedStatement preparedStatement = derbyEnvironment.getConnection().prepareStatement(sql);
8         preparedStatement.setString(1, statName);
9         preparedStatement.setTimestamp(2, new Timestamp(msFrom));
10        preparedStatement.setTimestamp(3, new Timestamp(msTo));
11
12        List<Statistics> statList = new ArrayList<Statistics>();
13        ResultSet rs = preparedStatement.executeQuery();
14        while (rs.next()) {
15            statList.add(buildNewStat(rs));
16        }
17
18        return statList;
19    }
20
21    public Statistics buildNewStat(ResultSet rs) {
22        Statistics stat = new Statistics();
23        stat.setStatisticID(rs.getLong("StatisticID"));
24        stat.setStatisticName(rs.getString("StatisticName"));
25        stat.setStatisticValue(rs.getDouble("StatisticValue"));
26
27        return stat;
28    }
29}
```

javac MyDaoClass.java

## MyDaoClass.java

```
// Bytecode stream: 03 3b 84 00 01 1a 05
// 68 3b a7 ff f9
// Disassembly:

iconst_0      // 03
istore_0       // 3b
iinc 0, 1     // 84 00 01
iload_0        // 1a
iconst_2       // 05
imul           // 68
istore_0       // 3b
goto -7       // a7 ff f9
```

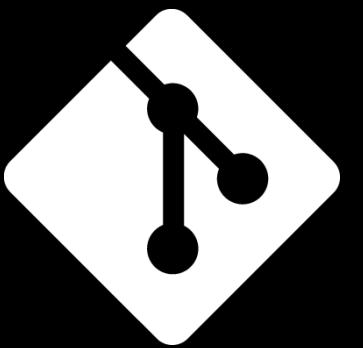


# Instruments

JDK



GIT



IntelliJ Idea



Gradle

