

Chapter 1

REPL

Listing 1.1: C++ code using listings

```
1  #include <stdio.h>
2  #include <stddef.h>
3  #include <stdint.h>
4  #include <stdlib.h>
5  #include <string>
6  #include <sstream>
7  #include <iostream>
8  #include <functional>
9  #include <tuple>
```

Listing 1.2: C++ code using listings

```
1  int main(int argc, char* argv[])
2  {
3      REPL repl;
4      repl.run([](auto str, auto& ss) {
5          if (str == ".exit") return -1;
6          else
7              {
8                  ss << "Unrecognized command ["
9                  << str << "]"";
10                 return 0;
11             }
12     });
13     exit(EXIT_SUCCESS);
14     return 0;
15 }
```

Listing 1.3: C++ code using listings

```

1  using PREPLCallback = int(
2      const std::string&,
3      std::stringstream& ss);
4  using REPLCallback = std::function<PREPLCallback>;
5  class REPL
6  {
7  public:
8      REPL() :
9          in(std::cin),
10         out(std::cout)
11     { }
12     REPL(std::istream& in, std::ostream& out) :
13         in(in),
14         out(out)
15     { }
16
17     void run(REPLCallback f) { ... }
18 private:
19     std::istream& in;
20     std::ostream& out;
21
22     std::string read_input() { ... }
23 };

```

Listing 1.4: C++ code using listings

```

1  void run(REPLCallback f)
2  {
3      std::stringstream ss;
4      while (true)
5      {
6          out << "db> ";
7
8          ss.str("");
9          ss.clear();
10
11         auto str = read_input();
12         auto status = f(str, ss);
13         if (status > 0)
14             return exit(status);
15         else if (status < 0)
16             break;
17
18         out << ss.str() << std::endl;
19     }
20 }

```

Listing 1.5: C++ code using listings

```
1  std::string read_input()
2  {
3      std::string str;
4      std::getline(in, str);
5      if (str.size() <= 0)
6      {
7          out << "Error reading input." << std::endl;
8          exit(EXIT_FAILURE);
9      }
10     return str;
11 }
```
