Source code:

```
#include <iostream>
#include <string>
#include <vector>
#include <ctime>
#include <cstdlib>
#include <algorithm>
#include <fstream>
#include <sstream>
#include <cctype>
#include <set>

class Hangman {
public:
   Hangman(std::string level) {
      srand(time(NULL));
      if (level == "easy") {
         words_and_hints_ = {{"cat", "A common household pet"},
                             {"dog", "A common household pet"},
                             {"elephant", "A large mammal with a long trunk"},
                             {"giraffe", "A tall giraffe-like mammal"},
                             {"lion", "A large African cat"},
                             {"tiger", "A large Asian cat"}};
      } else if (level == "intermediate") {
         words_and_hints_ = {{"rose", "A type of flower"},
                             {"sunflower", "A type of flower"},
                             {"tulip", "A type of flower"},
                             {"daisy", "A type of flower"},
                             {"orchid", "A type of flower"},
```

```
                {"lily", "A type of flower"}};
    } else if (level == "hard") {
        words_and_hints_ = {{"unitedstates", "A country in North America"},
                {"brazil", "A country in South America"},
                {"russia", "A country in Europe"},
                {"india", "A country in South Asia"},
                {"china", "A country in East Asia"},
                {"australia", "A country in Oceania"}};
    }

    if (words_and_hints_.empty()) {
        throw std::invalid_argument("Invalid level: no words and hints provided");
    }
    reset();
}

void guess(char c) {
    // Convert character to lowercase
    c = std::tolower(c);

    // Check if the character has already been guessed
    if (guessedLetters_.find(c) != guessedLetters_.end()) {
        std::cout << "You've already guessed '" << c << "'. Try a different letter.\n";
        return;
    }

    bool correctGuess = false;
    for (size_t i = 0; i < word_.length(); i++) {
        if (std::tolower(word_[i]) == c) {
```

```
            display_[i] = word_[i];
            correctGuess = true;
        }
    }
    if (!correctGuess) {
        --lives_;
    }
    guessedLetters_.insert(c); // Record the guessed letter
}

bool isOver() const {
    return lives_ == 0 || display_ == word_;
}

void print() const {
    std::cout << "Word: " << display_ << "\n";
    std::cout << "Hint: " << hint_ << "\n";
    std::cout << "Lives: " << lives_ << "\n";
    std::cout<<"Number of letters in the word:"<<word_.length()<<"\n";

    // Print hangman figure
    std::cout << "   _________\n";
    std::cout << "  |        |\n";
    std::cout << "  |        ";
    if (lives_ < 6) std::cout << "O";
    std::cout << "\n";
    std::cout << "  |       ";
    if (lives_ < 5) std::cout << "/";
    if (lives_ < 4) std::cout << "|";
```

```
        if (lives_ < 3) std::cout << "\\";
        std::cout << "\n";
        std::cout << "  |      ";
        if (lives_ < 2) std::cout << "/";
        if (lives_ < 1) std::cout << " \\";
        std::cout << "\n";
        std::cout << "__|__\n";
        std::cout << "\n";
    }

    void reset() {
        guessedLetters_.clear();
        int index = rand() % words_and_hints_.size();
        word_ = words_and_hints_[index].first;
        hint_ = words_and_hints_[index].second;
        numLetters_ = word_.length();
        display_ = std::string(numLetters_, '_');
        lives_ = 6;
    }

public:
    std::vector<std::pair<std::string, std::string>> words_and_hints_;
    std::string word_;
    std::string hint_;
    std::string display_;
    std::set<char> guessedLetters_; // To store guessed letters
    int numLetters_;
    int lives_;
};
```

```
int main() {
    std::cout << "****************************************\n";
    std::cout << "* Welcome to Hangman! The word guessing *\n";
    std::cout << "* game where you have to guess a word *\n";
    std::cout << "* letter by letter before you run out        *\n";
    std::cout << "* of lives. Good luck!       *\n";
    std::cout << "****************************************\n";

    std::string level;
    std::cout << "Choose a level (easy, intermediate, hard): ";
    std::cin >> level;

    do {
        Hangman game(level);

        while (true) {
            game.print();
            std::cout << "Enter a guess: ";
            char guess;
            std::cin >> guess;

            try {
                game.guess(guess);
            } catch (const std::exception& e) {
                std::cerr << "Invalid input: " << e.what() << "\n";
                continue;
            }
```

```
            if (game.isOver()) {
                break;
            }
        }

        game.print();

        if (game.isOver() && game.display_ == game.word_) {
            std::cout << "Congratulations, you won!\n";
        } else {
            std::cout << "Sorry, you lost. The word was: " << game.word_ << "\n";
        }

        std::cout << "Do you want to play again? (yes/no): ";
        std::string playAgain;
        std::cin >> playAgain;

        if (playAgain != "yes") {
            break;
        }
    } while (true);

    return 0;
}
```