```
# Project :- Hangman (Simple Game)

import random
def choose_word(word_list):
    return random.choice(word_list).lower()

def display_word(word, guessed_letters):
    return "".join([letter if letter in guessed_letters else "_" for letter in word])

def main():
    print("Welcome to Guess the Word!")
    # This is the list of words that will be used
    word_list = [ "accuracy", "algorithm", "allocate", "analog", "antivirus", "archive",
    "array", "backup", "bandwidth", "binary", "bit", "boolean", "browser",
    "buffer", "bug", "cache", "callback", "captcha", "character", "charset",
    "cipher", "client", "clipboard", "codec", "command", "compiler",
    "compression", "constructor", "cookie", "cursor", "database", "debug",
    "decompiler", "default", "delimiter", "deprecated", "deterministic",
    "device", "dialog", "directory", "driver", "emulator", "encryption",
    "endian", "error", "exception", "executable", "extension", "firewall",
    "firmware", "framework", "freeware", "function", "gateway", "handler",
    "hardware", "hexadecimal", "host", "hyperlink", "identifier", "index",
    "inheritance", "instance", "interface", "interpreter", "iteration",
    "library", "linker", "localhost", "loop", "malware", "method", "module",
    "multithreading", "namespace", "network", "null", "object", "operator",
    "overload", "override", "package", "parameter", "pixel", "pointer",
    "polymorphism", "port", "protocol", "proxy", "query", "queue",
    "recursion", "regular", "expression", "repository", "sdk", "server",
    "software", "stack", "syntax", "token", "virtual", "widget" ]

    word = choose_word(word_list)
    guessed_letters = set()
    attempts = 0
    # There should be max limit to attempt otherwise it will be impossible to lose
    max_attempts = 10
    print(f"The word has {len(word)} letters.")

    while attempts < max_attempts:
        print(f"\nWord: {display_word(word, guessed_letters)}")
        guess = input("Enter your guess (single letter): ").lower()
        if len(guess) != 1 or not guess.isalpha():
            print("Invalid input. Please enter a single letter.")
            continue

        if guess in guessed_letters:
            print("You already guessed that letter.")
        elif guess in word:
            guessed_letters.add(guess)
            print("Correct!")
            if set(word) == guessed_letters:
                print(f"\nCongratulations! You've guessed the word: {word}")
                break
        else:
            attempts += 1
            print(f"Incorrect! You have {max_attempts - attempts} attempts left.")
    if attempts >= max_attempts:
        print(f"\nGame Over. The word was: {word}")

if __name__ == "__main__":
    main()
```