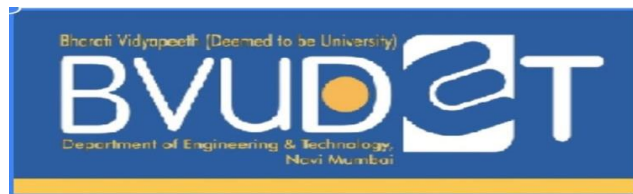


**BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY)**

**COLLEGE OF  
ENGINEERING**



**DEPARTMENT OF ENGINEERING & TECHNOLOGY OFF CAMPUS,  
KHARGHAR, NAVI MUMBAI, 410210**



## **Mini Project Report**

**On**

### **Hangman Game**

**Subject:- - Python Programming**

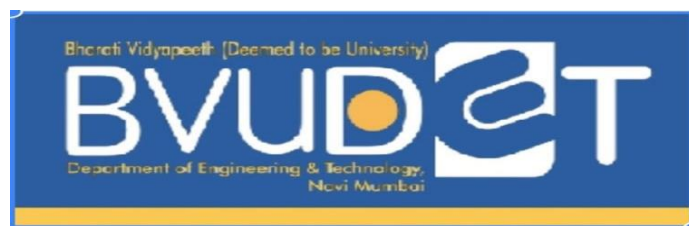
*Presented By*

<b>Sr No.</b>	<b>Name</b>	<b>PRN</b>
1.	Srijan Sundaram	2343110291
2.	Siddhant Sinha	2343110098

**Signature of Internal Examiner**

**Signature of HOD**

**BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY)**  
**COLLEGE OF  
ENGINEERING**  
**DEPARTMENT OF ENGINEERING & TECHNOLOGY OFFCAMPUS,  
KHARGHAR, NAVI MUMBAI, 410210**



**CERTIFICATE**

This is to certify that the requirements for the project report entitled 'Hangman Game' have been successfully completed by the following students:

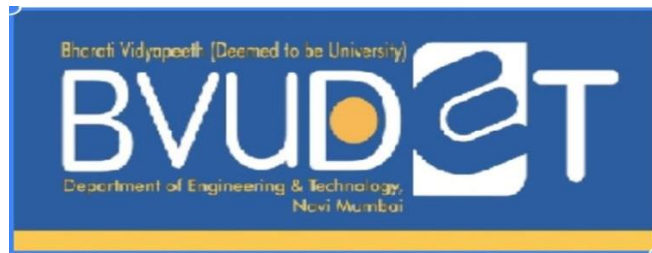
<b>Name</b>	<b>PRN No.</b>
1.Srijan Sundaram	2343110291
2.Siddhant Sinha	2343110098

in partial fulfillment of B.Tech in the Department of CSE AIML, BVU DET, during the Academic Year 2024 – 2025.

**Subject In charge**  
**(Prof. Rupali Sharma)**

**BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY)**

**DEPARTMENT OF ENGINEERING & TECHNOLOGY OFFCAMPUS,  
KHARGHAR, NAVI MUMBAI, 410210**



**DECLARATION**

We declare that this written submission for B.TECH project entitled “Hangman Game” represent our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission have not been taken when needed.

**Project Group Members**

**Signature**

Srijan Sundaram

\_\_\_\_\_

Siddhant Sinha

\_\_\_\_\_

## **Abstract**

This is a simple Hangman game using Python programming language. We can use this as a small project to boost their programming skills and understanding logic. The Hangman program randomly selects a secret word from a list of secret words. The random module will provide this ability, so line 1 in program imports it. Hangman is a popular word game in which one player (the "chooser") chooses a secret word and another player (the "guesser") attempts to guess the word one letter at a time. If a guessed letter appears in the word, all instances of it are revealed. If not, the guesser loses a chance. If the guesser figures out the secret word before he or she runs out of chances, he or she wins. If not, the player who chose the word wins.

# **Index**

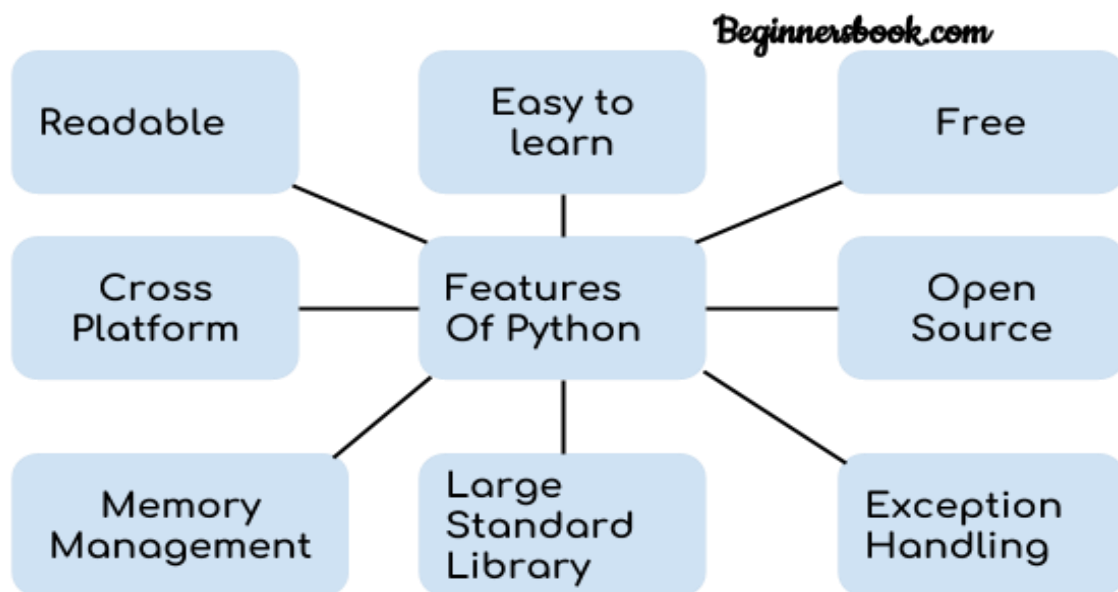
<b>Sr No.</b>	<b>Title</b>	<b>Page No.</b>
1	Introduction	6-8
2	Definition	9
3	Application	10
4	Analysis	11-12
5	Implementations	13-16
6	Results	17
7	Conclusion	18
8	Reference	19

# Introduction

## Python Programming language

Python is one of the many open source object oriented programming application software available in the market . Python is developed by **Guido van Rossum**. Guido van Rossum started implementing Python in 1989. Python is a very simple programming language so even if you are new to programming, you can learn python without facing any issues. Some of the many uses of Python are application development, implementation of automation testing process, allows multiple programming build, fully constructed programming library, can be used in all the major operating systems and platforms, database system accessibility, simple and readable code, easy to apply on complex software development processes, aids in test driven software application development approach, machine learning/ data analytics, helps pattern recognitions, supported in multiple tools, permitted by many of the provisioned frameworks, etc.

Some features of Python are-



1. **Readable:** Python is a very readable language.
2. **Easy to Learn:** Learning python is easy as this is a expressive and high level programming language, which means it is easy to understand the language and thus easy to learn.
3. **Cross platform:** Python is available and can run on various operating systems such as Mac, Windows, Linux, Unix etc. This makes it a cross platform and portable language.
4. **Open Source:** Python is a open source programming language.
5. **Large standard library:** Python comes with a large standard library that has some handy codes and functions which we can use while writing code in Python.
6. **Free:** Python is free to download and use. This means you can download it for free and use it in your application. See: [Open Source Python License](#). Python is an example of a FLOSS (Free/Libre Open Source Software), which means you can freely distribute copies of this software, read its source code and modify it.
7. **Supports exception handling:** If you are new, you may wonder what is an exception? An exception is an event that can occur during program exception and can disrupt the normal flow of program. Python supports exception handling which means we can write less error prone code and can test various scenarios that can cause an exception later on.
8. **Advanced features:** Supports generators and list comprehensions. We will cover these features later.
9. **Automatic memory management:** Python supports automatic memory management which means the memory is cleared and freed automatically

### Applications of Python programming language

Python can be used to develop different applications like web applications, graphic user interface based applications, software development application, scientific and numeric applications, network programming, Games and 3D applications and other business applications. It makes an interactive interface and easy development of applications. You may be wondering what all are the applications of Python. There are so many applications of Python, here are some of the them.

1. **Web development** – Web framework like Django and Flask are based on Python. They help you write server side code which helps you manage database, write backend programming logic, mapping urls etc.
2. **Machine learning** – There are many machine learning applications written in Python. Machine learning is a way to write a logic so that a machine can learn and solve a particular problem on its own. For example, products recommendation in websites like Amazon, Flipkart, eBay etc. is a machine learning algorithm that recognises user's interest. Face recognition and Voice recognition in your phone is another example of machine learning.
3. **Data Analysis** – Data analysis and data visualisation in form of charts can also be developed using Python.
4. **Scripting** – Scripting is writing small programs to automate simple tasks such as sending automated response emails etc. Such type of applications can also be written in Python programming language.

5. **Game development** – You can develop games using Python.
6. You can develop **Embedded applications** in Python.
7. **Desktop applications** – You can develop desktop application in Python using library like TKinter or QT.

## Introduction to the game

The **Hangman game** is a simple and classic word-guessing game that involves one player thinking of a word or phrase and the other trying to guess it by suggesting letters. The game is often used to test vocabulary and spelling skills, and it can be played on paper, a whiteboard, or digitally.

### Key elements of Hangman:

1. **Secret Word or Phrase:** One player selects a word (or phrase) and keeps it hidden. It is typically represented by a series of blank spaces, each blank representing a letter of the word.
2. **Guesses:** The other player guesses letters one at a time. If the guessed letter is in the word, it is revealed in the appropriate blank space(s). If not, it counts as a wrong guess.
3. **Drawing the Hangman:** For each wrong guess, a part of a stick figure ("the hangman") is drawn. This figure typically consists of:
  - Head
  - Body
  - Two arms
  - Two legs
4. **Winning or Losing:** The player wins by guessing all the letters in the word before the full figure of the hangman is drawn. They lose if the full hangman is drawn without the word being guessed.

### Example of Gameplay:

- The word is "apple," represented as \_ \_ \_ \_ \_.
- The player guesses "e," which is correct. The word becomes \_ \_ \_ \_ e.
- The player then guesses "z," which is incorrect. A part of the hangman is drawn.
- The process continues until the word is either guessed or the full figure is drawn.

Hangman is great for improving language skills while being fun and engaging.



## Definition

The **Hangman game** is a word-guessing game in which one player thinks of a word, phrase, or sentence, and the other player tries to guess it by suggesting individual letters. For each incorrect letter guessed, a part of a stick-figure drawing of a "hangman" is added. The goal of the guessing player is to figure out the word before the drawing of the hangman is completed, typically in six or seven incorrect guesses. If the full figure is drawn before the word is guessed, the game is lost. If the word is guessed, the guessing player wins.

# Application

The **Hangman game** has various applications beyond its entertainment value, making it a versatile tool for both education and recreation. Some key applications include:

## **1. Educational Tool:**

- **Vocabulary Building:** Hangman is widely used in language learning to help students improve their vocabulary and spelling skills. By guessing letters, players become familiar with word structures and new terms.
- **Reinforcing Spelling Rules:** It helps learners practice common spelling patterns, making it useful in English as well as other language classes.
- **Foreign Language Learning:** It can be used to teach non-native speakers new words in a foreign language, helping them memorize and use words in a fun and engaging way.

## **2. Critical Thinking and Problem Solving:**

- Players develop strategies for guessing words and letters, enhancing their problem-solving and pattern recognition skills.
- It encourages logical thinking, as players must deduce the possible word based on the structure and the letters already revealed.

## **3. Icebreaker or Group Activity:**

- Hangman can be used as a social game or group activity in classrooms, family gatherings, or parties, serving as an icebreaker or collaborative task that encourages teamwork and communication.

## **4. Cognitive Development:**

- The game aids in improving memory and concentration, especially in young learners, by requiring them to remember and use previously guessed letters effectively.

## **5. Recreational Fun:**

- Hangman is commonly played for leisure, both online and offline, offering a simple and engaging way to pass time for players of all ages.

## **6. Digital and AI Applications:**

- Modern digital versions of Hangman incorporate AI, adaptive difficulty levels, and a wide range of vocabulary to enhance gameplay, making it accessible for a broader audience online.

These applications make Hangman a flexible game that combines fun with learning, applicable in various educational, social, and cognitive contexts.

# Analysis

The **Hangman game** offers a rich basis for analysis, encompassing aspects of game design, learning theory, and strategy. Below is a breakdown of key analytical perspectives:

## 1. Game Mechanics:

- **Core Gameplay:** Hangman is a turn-based game where players alternate between guessing letters and receiving feedback (correct/incorrect). The game involves elements of randomness (the word chosen by one player) and skill (the strategic guessing of letters).
- **Feedback Loop:** Each guess provides immediate feedback—correct guesses reveal part of the word, while incorrect ones contribute to drawing the hangman. This feedback loop is essential for engaging the player and guiding their next guesses.

## 2. Probability and Strategy:

- **Letter Frequency:** Effective players often rely on letter frequency data, focusing first on common letters (e.g., vowels like 'e,' 'a,' 'o') to maximize their chances of quickly revealing key parts of the word.
- **Pattern Recognition:** Once a few letters are revealed, the player shifts to recognizing possible word structures and patterns, narrowing down the set of potential words.
- **Risk Management:** Players must balance risky guesses with safer options. For example, guessing an uncommon letter may provide more insight if correct, but it also increases the risk of a penalty if wrong.

## 3. Cognitive and Learning Impact:

- **Memory and Recall:** The game requires players to remember previous guesses and use this information to avoid repeating mistakes, thus reinforcing memory skills.
- **Spelling and Vocabulary Development:** Hangman reinforces correct spelling patterns and helps players practice new vocabulary. It can also highlight letter combinations common in a language.
- **Deductive Reasoning:** Players use deductive reasoning to eliminate impossible letter combinations and focus on the most probable word candidates.

## 4. Game Theory:

- **Optimal Play:** In theory, the optimal strategy would involve guessing letters in descending order of frequency in the target language. However, this changes dynamically as letters are revealed or eliminated, so flexibility is key.
- **Game Balance:** The game is balanced by the limited number of wrong guesses allowed. While a word can be chosen to be relatively easy or difficult, the guess limit introduces a clear challenge for players, ensuring that the game is neither too easy nor too hard.

## 5. Psychological Aspects:

- **Suspense and Motivation:** The gradual drawing of the hangman figure creates suspense, motivating players to avoid incorrect guesses. This element of tension is what keeps the game engaging.
- **Failure vs. Reward:** The visual representation of failure (the hangman figure) contrasts with the reward of guessing the word, creating a strong emotional impact. Players experience a combination of relief and satisfaction when they succeed, and disappointment if they fail.

## 6. Limitations and Criticisms:

- **Word Length and Difficulty:** The difficulty of the game is largely determined by the length and complexity of the chosen word. Short words or obscure vocabulary can make the game either too easy or frustrating.
- **Cultural Bias:** Words chosen for the game may be influenced by cultural or linguistic bias. For example, a word commonly known in one culture might be unfamiliar in another, leading to unequal playing fields.
- **Finite Guessing Pool:** Since the number of incorrect guesses is typically limited to six or seven, players may feel restricted, especially when facing complex words.

## 7. Digital Evolution:

- **Adaptive Learning:** In digital versions of the game, machine learning algorithms can adjust the difficulty level based on player performance, making the game more personalized and educational over time.
- **Interactivity:** Online and app-based versions offer features like hints, timers, and levels, adding layers of complexity and replayability that extend the classic game's appeal.

## Conclusion:

The Hangman game is simple in design but deep in its cognitive and strategic elements. While primarily a game of guessing, it engages higher-level thinking processes, including memory, pattern recognition, and logical deduction. Its adaptability for various educational and recreational contexts keeps it relevant across different environments, even as it has evolved in the digital age.

# Implementation

## Code

```
import random

def get_random_word():
    words = ['python', 'hangman', 'challenge', 'programming', 'developer']
    return random.choice(words)

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

def hangman():
    word = get_random_word()
    guessed_letters = set()
    tries = 6
    guessed_word = display_word(word, guessed_letters)
    name = input("Enter your name: ")

    print(f"{name}, Welcome to Hangman!")
    print(guessed_word)

    while tries > 0 and ' _' in guessed_word:
        guess = input("Guess a letter: ").lower()

        if len(guess) != 1 or not guessed_word:
            print("Invalid Input. Please enter a single letter.")
            continue

        if guess in guessed_letters:
            print(f"You already guessed the letter {guess}. Try another one!")
            continue

        guessed_letters.add(guess)

        if guess in word:
            print(f"Good job, {name}! You guessed the letter {guess}!")
        else:
            print("Incorrect Guess. Try again.")
            tries -= 1

        guessed_word = display_word(word, guessed_letters)
        print(guessed_word)
        print(f"Remaining attempts: {tries}")
```

```

if '_' not in guessed_word:
    print(f"Congratulations {name}! You have won!")
else:
    print(f"Game Over! The word was {word}.")

if __name__ == "__main__":
    hangman()

```

## Output

```

PS C:\Users\asus> python -u "C:\Users\asus\AppData\Local\Temp\tempCodeRunnerFile.python"
Enter your name: siddhant
siddhant, Welcome to Hangman!

_ _ _ _ _
Guess a letter: g
Good job, siddhant! You guessed the letter g!
_ _ _ g _ _ _
Remaining attempts: 6
Guess a letter: h
Good job, siddhant! You guessed the letter h!
h _ _ g _ _ _
Remaining attempts: 6
Guess a letter: a
Good job, siddhant! You guessed the letter a!
h a _ g _ a _
Remaining attempts: 6
Guess a letter: n
Good job, siddhant! You guessed the letter n!
Remaining attempts: 6
Guess a letter: n
Remaining attempts: 6
Remaining attempts: 6
Guess a letter: n
Good job, siddhant! You guessed the letter n!
h a n g _ a n
Remaining attempts: 6
Guess a letter: m
Good job, siddhant! You guessed the letter m!
h a n g m a n
Remaining attempts: 6
Congratulations siddhant! You have won!

```

## For Invalid attempts

```
Enter your name: siddhant
siddhant, welcome to Hangman!

_ _ _ _ _
Guess a letter: h
Incorrect Guess. Try again.

_ _ _ _ _
Remaining attempts: 5
Guess a letter: y
Incorrect Guess. Try again.

_ _ _ _ _
Remaining attempts: 4
Guess a letter: n
Incorrect Guess. Try again.

_ _ _ _ _
Remaining attempts: 3
Guess a letter: a
Incorrect Guess. Try again.

_ _ _ _ _
Remaining attempts: 2
Guess a letter: o
Good job, siddhant! You guessed the letter o!

_ _ _ _ _
Remaining attempts: 2
Guess a letter: u
Incorrect Guess. Try again.

_ _ _ _ _
Remaining attempts: 1
Guess a letter: k
Incorrect Guess. Try again.

_ _ _ _ _
Remaining attempts: 0
Game Over! The word was developer.
```

**Invalid attempt for putting multiple letters.**

```
Enter your name: sid
sid, welcome to Hangman!

_ _ _ _ _
Guess a letter: hi
Invalid Input. Please enter a single letter.
Guess a letter: █
```



## Results

The **Hangman game** demonstrates a balance between entertainment and educational value, offering cognitive benefits through vocabulary building, memory reinforcement, and logical reasoning. The game's simple mechanics—where players guess letters to uncover a hidden word while avoiding too many incorrect guesses—make it accessible to a wide audience while fostering critical thinking and pattern recognition.

Strategically, players must leverage common letter frequencies and deductive reasoning to minimize mistakes and reveal the word before time runs out. The game encourages players to analyze word structures, improve spelling, and apply probability in choosing their guesses. However, factors like word difficulty and cultural bias may influence the fairness and challenge level of the game.

In educational contexts, Hangman is an effective tool for language learners, providing a fun and engaging method to enhance vocabulary retention and spelling skills. Meanwhile, the game's suspenseful mechanics, combined with the visual representation of failure, create a motivating and emotionally engaging experience.

Overall, Hangman remains a timeless and adaptable game, with applications ranging from educational reinforcement to recreational play. Its continued popularity, especially in digital formats, underscores its lasting appeal as both a learning tool and a source of enjoyment.

## Conclusion

The **Hangman game** is a classic word-guessing game that combines entertainment with educational benefits. Its simple yet engaging mechanics, which involve guessing letters to reveal a hidden word, make it accessible to players of all ages and skill levels. The game fosters cognitive development by promoting vocabulary learning, spelling improvement, memory retention, and logical reasoning.

As a versatile tool, Hangman can be used in classrooms for language instruction, as well as in social or recreational settings for fun and friendly competition. Despite its simplicity, the game challenges players to think strategically, balancing risk and reward with each guess. With its enduring appeal, Hangman continues to evolve in digital formats, enhancing its adaptability while maintaining the core elements that make it both fun and educational. This timeless game remains an effective way to engage minds and foster learning through playful interaction.

## **Reference**

1. Tutorial: Python Hangman Game Tutorial.
2. "Python Crash Course" by Eric Matthes.