

**VISVESWARAYA TECHNOLOGICAL UNIVERSITY**  
**Jnana Sangama,Belagavi-590018**



**MINI PROJECT REPORT ON**  
**Hangman**

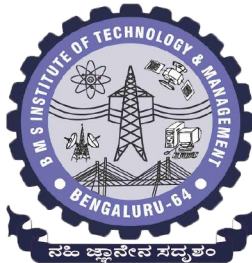
**Submitted By**

Arjit Praveen Kumar (1BY23CS030)  
Amogh Shankar (1BY23CS021)  
Bhagyadarshan Patil (1BY23CS038)

**Under The Guidance Of**

**Dr. Vidya R Pai**

**Department of CSE**

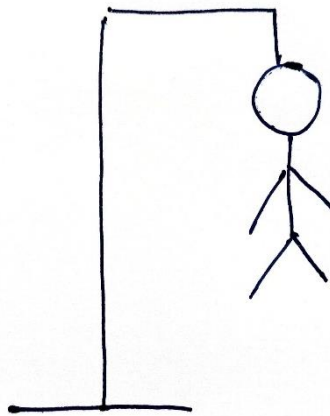


**BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT**  
**DEPARTMENT NAME**

**Avalahalli , Yelahanka , Bengaluru – 560064.**

# INTRODUCTION

*Hangman* is a guessing game involving 2 players, one of which chooses a word and gives a hint of what the word is to the other player, who should guess each letter of the words in a given attempt. A hanging stick figure determines the number of attempts, upon the completion of the stick figure, the guesser runs out of attempts.



## Setup

One player thinks of a word or phrase and writes down a series of blank spaces representing each letter of the word. For example, if the word is "apple", they would write " \_ \_ \_ \_ \_ ". A gallows is drawn on a piece of paper or a whiteboard, which will be used to draw parts of a stick figure with each incorrect guess.

## Gameplay

The other player guesses one letter at a time. If the guessed letter is in the word, the first player fills in the blanks with the correct letter. If the guessed letter is not in the word, the first player draws one part of the stick figure on the gallows. The figure consists of a head, body, 2 arms, and 2 legs – allowing 6 incorrect guesses before the game ends.

## Winning and Losing

The guessing player wins if they guess all the letters in the word before the stick figure is fully drawn. The first player wins if the stick figure is completely drawn before the words is guessed

To produce a hangman simulation using Python, it's imperative to understand what variables should be used and what their respective functions should be.

## PROBLEM DEFINITION

Our code will contain 4 separate sections. The following includes:

1. The function: This serves as a method to update and display our hangman figure with every increment of a variable, let's say 'count', where **count** is a variable that increments with every wrong answer. The variable passed into the function **struct()** uses nested IF statements to verify the count value and display the associated structure.

*Sample snippet:*

```
def struct(a):
    if a==0:
        print("+---+")
        print("|   |")
        print("   |")
        print("   |")
        print("   |")
        print("   |")
        print("   ==")

    if a==1:
        print(" +---+")
        print(" |   |")
        print(" O   |")
        print("     |")
        print("     |")
        print("     |")
        print("     ==")
```

2. Initialization: The initialization is made after the function definition.

```
W = input("Enter the word: ") .upper()
n = len(W)
clue = input("Enter a clue related to the word: ")
lis, b, c = [], [], []
z, wg = 0, 0 #wg = wrong guesses counter
```

This is where the gamemaster inputs the word which has to be guessed correctly by the users.

3. The game: The 3<sup>rd</sup> section of the program consists of using a WHILE loop to initiate the game. In this loop, the user is asked for a letter, compared to a list of characters in the solution.
4. Output: The final section involves congratulating the victor if they have won the game

```
print(f" CONGRATULATIONS! The word is {' '.join(lis)}, You are the Victor.")
```

## PROBLEM SOLUTION

### The code:

```
import sys
def struct(a):
    if a==0:
        print("+---+")
        print("|   |")
        print("   |")
        print("   |")
        print("   |")
        print("   |")
        print("   ===")

    if a==1:
        print(" +---+")
        print(" |   |")
        print(" O   |")
        print("   |")
        print("   |")
        print("   |")
        print("   ===")

    if a==2:
        print(" +---+")
        print(" |   |")
        print(" O   |")
        print(" |   |")
        print("   |")
        print("   |")
        print("   ===")

    if a==3:
        print(" +---+")
        print(" |   |")
        print(" O   |")
        print(r" /|   |")
        print("   |")
        print("   |")
        print("   ===")

    if a==4:
        print(" +---+")
        print(" |   |")
        print(" O   |")
        print(r" /\   |")
```

```

        print("      |")
        print("      |")
        print("      ===")
    if a==5:
        print("  +---+")
        print("  |   |")
        print("  O   |")
        print(r" /|\  |")
        print("  |   |")
        print("      |")
        print("      ===")
    if a==6:
        print("  +---+")
        print("  |   |")
        print("  O   |")
        print(r" /|\  |")
        print("  |   |")
        print(r" /     |")
        print("      ===")
    if a==7:
        print("  +---+")
        print("  |   |")
        print("  O   |")
        print(r" /|\  |")
        print("  |   |")
        print(r" / \   |")
        print("      ===")

w=input("Enter the word: ").upper()
n=len(w)
clue=input("Enter a clue related to the word: ")
for i in range(0,51):
    print("  ")
lis, b, c =[], [], []
z, wg = 0, 0
struct(wg) #printing the initial figure
for i in w:
    lis.append(' ')
print(f"It is a {n} letter word.")
print('Clue:',clue)
for i in w:
    b.append(i)
    c.append(i)

```

```

while lis!=c:
    l=input('Guess a letter:').upper()
    z=0
    for i in b:
        if i==l:
            ix=b.index(i)
            lis[ix]=i
            b[ix]=' '
        else:
            z+=1
    print('\n'+ ' '.join(lis))
    print(n*'_ ', '\n')
    if z==n:
        wg+=1
        print('wrong guess!!!')
        struct(wg)
        print(f'LIVES LEFT: {7-wg}'+'\n')
    if wg==7:
        print(' Sorry the man has been HANGGED ;( ')
        print(' Better luck Next time !')
        sys.exit()
    else:
        print('Correct Guess, Right on!','\n')

print(f" CONGRATULATIONS! The word is {' '.join(lis)},You are the Victor.")
print("Thank You for playing.")

```

Notable concepts used in the program:

1. Functions: Function which serves as a *procedure* (no return statement)
2. Sys module: The sys module in Python provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter

This program makes use of the **sys.exit()** method. Where, **sys.exit([arg])** exits the Python interpreter. The optional argument can be an integer giving the exit status (defaulting to zero) or another type to print an error message.

3. In-built functions (string & array manipulation): **.upper()**, **len()**, **.append()**
4. Conditional statements: IF ELSE statement used to compare values using the *compare operator* '=='
5. Loops: WHILE loop is a *precondition* loop which verifies the condition at the start of the loop. And accordingly, carry out the following block of code if the condition is satisfied.

**Output:**

It is a 3 letter word.

Clue: Feline

Guess a letter:C

C

— — —

Correct Guess, Right on!

Guess a letter:a

C A

— — —

Correct Guess, Right on!

Guess a letter:t

CAT



Correct Guess, Right on!

CONGRATULATIONS! The word is C A T, You are the Victor.  
Thank You for playing.

## **BENEFITS**

Hangman is a game widely popular amongst youth. The benefits of playing hangman involve;

- **Vocabulary Building**

Hangman encourages players to think of different words, helping to expand and reinforce vocabulary. It can introduce new words and help players remember spelling and meaning through repeated exposure.

- **Critical Thinking and Problem-Solving:**

Players need to use logic and deductive reasoning to guess letters and figure out the word before running out of chances. This enhances their critical thinking and problem-solving skills.

- **Educational Fun:**

Hangman is an enjoyable and engaging way to learn, making it easier for both children and adults to absorb and retain information. It turns learning into a game, which can be particularly effective in educational settings.

- **Stress Relief and Relaxation:**

Playing games like Hangman can provide a mental break from stress and anxiety. Engaging in a fun and light-hearted activity helps to relax the mind and can improve overall mood and mental well-being.

- **Boosting Memory and Cognitive Function:**

Hangman challenges players to recall words and letters from memory, which exercises and strengthens memory functions. Regularly playing such word games can enhance cognitive abilities, including memory retention and recall speed.



# CONCLUSION

*What is Hangman, it's benefits to our linguistic and psychological development?*

Hangman is a game that involves 2 parties: the gamemaster and the guessers. A word is given by the game master which has to be guessed by the guessers. A series of 'dashes' are drawn next to a figure representing an empty gallows. The number of dashes corresponds to the length of the word. The guesser then takes turns, guessing letters in the word according to the given clue while considering probabilistic parameters. If the player guesses wrong, a stick figure representing a human is drawn and hung on the gallows. The stick figure is drawn in a series of 6 distinct parts: the head, the body, 2 arms, and 2 legs. If the figure is complete, the guesser loses. Unless the given word is spelled correctly.

Hangman plays a crucial role in our linguistic and psychological development. To begin with, it encourages us to think of different words, this helps in expanding and reinforcing our vocabulary. Furthermore, this allows us to retain foreign words that may have been introduced to the players during the game. Secondly, hangman focuses on players using logical and deductive reasoning to guess letters and figure out the word. Moreover, Hangman can provide a mental break from stress and anxiety and more importantly, serve as a serotonin-inducing activity. Finally, Hangman enhances our memory and cognitive function.

*How can Hangman be implemented in Python? What outputs can we expect? And what logic is applied?*

There are numerous ways of implementing a hangman game in Python. One such way is by using an installed module known as **Pygame** which involves setting up a "window" and using a clock and a loop to initiate the game. Moreover, it tracks the user's cursor and clock using event functions. However, a much more simple take can be implemented.

Firstly, the preinstalled module **sys** is imported. This allows us to exit the program when the guesser has run out of chances. Then, the function **struct()** is defined. This serves a method to print out gallows and stick figure, allowing the guesser to check how many more chances they have left. Thirdly, initializations are made, this involves: accepting an input from the gamemaster with the word to be guessed, and standard variable initialization which proves important in later stages of the program. Next, a while loop is created, within this loop is a block of code that allows the user to enter a letter that is compared against a letter already contained in the word provided by the game master. Lastly, a print statement using a formatting operator is provided to output the solution, and a prompt which congratulates the user if they have won the game.