In [ ]:

```
"""
file: adventure_game.pdf

Objective:

This project is a simple text associated adventure game generated in Python
using VS Code and GitHub Copilot.

My goal is to show basic Python concepts such as:

Functions related programming

Variables

User input handling

if/elif/else conditions

Loops used for replaying the game

Command line interface interaction

Players choose the role of an adventurer seeking treasure. They must choose
either forest or cave path and then take

decisions which reflect whether they win or lose their journey. The game
continuously takes loops until the player

wants to exit.

"""
```

In [ ]:

```
"""

Create the forest path

Actions:

Define the function forest_path() that describes the forest scenario

Provide the player with choices (follow a river or climb a tree)

Use an if-else structure to handle player choices

Use GitHub Copilot to generate the function body.
```

```
"""

# Define orest_path(player_name)
def forest_path(player_name):
    # print
    print(f"\n{player_name}, you are in the forest surrounded by towering trees.")

    # Ask a question to the player
    player_choice = input("Do you want to follow the river or climb a tree? (river/tree): ").strip().lower()

    # is this (player_choice == "river":) true? if yes, go through it only.
    if player_choice == "river":
        print("You follow the river and find a beautiful waterfall.")

    # is this (player_choice == "tree":) true? if yes, go through it only (not if)
    elif player_choice == "tree":
        print("You climb the tree and discover a beautiful view of the entire forest.")

    # When if and elif are not true, you can go into else option.
    else:
        print("Invalid choice. Please choose either'river' or 'tree' You get lost in the forest.")
```

In [ ]:

```
"""

Create the cave path

Actions:

Define the function cave_path() that describes the cave scenario

Provide the player with choices (light a torch or proceed in the dark)

Use conditionals to determine the outcome

Use GitHub Copilot to generate the function body.

"""

# Define cave_path(player_name)
def cave_path(player_name):
    # print
    print(f"\n{player_name}, you are in the cave.")

    # Ask a question to the player
    player_choice  = input("Do you want to light a torch or proceed in the dark? (torch/dark): ").strip().lower()

    # is this (player_choice == 'torch':) true? if yes, go through it only.
    if player_choice == 'torch':
        print("You light a torch and discover ancient cave paintings on the walls!.")

    # is this (player_choice == "dark":) true? if yes, go through it only.
    elif player_choice == "dark":
        print("You proceed in the dark and fall into a hidden pit. The adventure ends here.")

    # When if and elif are not true, you can go into else option.
    else:
        print("Invalid choice. You get lost in the cave.")
```

In [ ]:

```
"""

Run the adventure game

Call start_game() to begin the adventure

Ensure the program runs in a loop until the player completes their journey

Provide an option to restart the game after completion

Use GitHub Copilot to generate the function body.

"""


# Function to start the game, includes loop and replay

# Define start_game() with loop and replay featurers.
def start_game():

    # game runs through a loop until the player shows to replay.
    while True:
        # print
        print("\nWelcome to the Adventure Game!")

        # Ask player name
        player_name = input("What is your name, brave adventurer? ")

        # Start quest
        print(f"Hello, {player_name}! Your quest begins now.")

        # Outline the situation
        print("You find yourself at a crossroads.")
        print("Do you want to explore the forest or the cave?")

        # Ask the player for choosing an option
        player_choice = input("Type 'forest' or 'cave': ").strip().lower()
```

```python
        # Go to forest_path(player_name) already defined when
        # player_choice == "forest":
        if  player_choice == "forest":
            forest_path(player_name)

        # Go to cave_path(player_name) already defined when
        #  player_choice == "cave":

        elif  player_choice == "cave":
            cave_path(player_name)

        # When if and elif do not go through, go into the else option.
        else:
            print("Invalid choice. The adventure ends here.")

        # Ask the player if he or she wants to replay.

        restart = input("\nDo you want to play again? (yes/no): ").strip().lower()

        if restart != 'yes':
            print("Thank you for playing! Goodbye!")
            break    # break loop.

        else:
            print("\nRestarting your adventure...\n")

# Entry point

# these 2 lines codes support to this runs the game when the file is executed directly
if __name__ == "__main__":
    start_game()
```

In [ ]:

```python
"""
Results:

Welcome to the Adventure Game!

What is your name, brave adventurer? Gobind

Hello, Gobind! Your quest begins now.

You find yourself at a crossroads.

Do you want to explore the forest or the cave?
```

```
Type 'forest' or 'cave': cave

Gobind, you are in the cave.

Do you want to light a torch or proceed in the dark? (torch/dark): dark

You proceed in the dark and fall into a hidden pit. The adventure ends here.

Do you want to play again? (yes/no): no

Thank you for playing! Goodbye!

The game successfully shows Python basics, command line interface interactive input, and replay functions that cover
the project goals.

"""
```