

2303A10H5

BATCH-30

## **ASSIGNEMENT-04**

## PROBLEM:

Develop a basic ERC-20 token smart contract using Solidity that allows users to:

- Define token details such as name, symbol, decimals, and total supply
  - Transfer tokens between Ethereum accounts
  - Store and manage user token balances
  - Emit events to record token transfer activities on the blockchain

This practical helps understand state variables, mappings, constructors, events, and functions in Solidity, as well as the basics of Ethereum token standards.

The screenshot shows a code editor interface with the following details:

- File Structure:** The left sidebar shows a file tree with files like `ASSIGNMENT-3.docx`, `ASSIGNMENT-3.pdf`, `Simple_storage_gui.py`, and `wallet_interaction.py`.
- Code Editor:** The main area displays a Python script (`Ass 4.py`) and a Solidity contract (`ERC20Token`). The Python code uses `tkinter` for a GUI, `web3` for Ethereum interaction, and `solcx` for Solidity compilation.
- Solidity Contract (`ERC20Token`):**

```
contract ERC20Token {
    string public name;
    string public symbol;
    uint8 public decimals = 18;
    uint public totalSupply;

    mapping(address => uint) public balanceOf;

    event Transfer(address indexed from, address indexed to, uint value);

    constructor(string memory _name, string memory _symbol, uint _supply) {
        name = _name;
        symbol = _symbol;
        totalSupply = _supply * (10 ** uint(decimals));
        balanceOf[msg.sender] = totalSupply;
    }
}

```
compiled = compile_source(erc20_source)
_, contract_interface = compiled.popitem()
```
}
```
- Terminal:** The bottom terminal window shows the command line output of the Python script, which includes the path to the Solidity contract and its compilation.
- Bottom Bar:** The bottom bar includes icons for outline, timeline, and search, along with a status bar indicating "Air Moderate".

A screenshot of a Windows desktop environment. In the foreground, a code editor window titled "Ass 4.py" is open, displaying Python code for deploying an ERC20 token. The code uses the web3 library to interact with a Ganache instance. It includes functions for deploying tokens with specific names, symbols, and supplies, and handles exceptions for missing input fields. Below the code editor, the taskbar displays several pinned icons for applications like File Explorer, Microsoft Edge, and various Microsoft Office tools.

```
File Edit Selection View Go Run Terminal Help ← → 🔍 Block chain
```

```
EXPLORER BLOCK CHAIN Ass 4.py Ass 4.py ... Ass 4.py -
```

```
43     def deploy_token():
44         if not connected:
45             messagebox.showerror("Error", "Ganache not connected")
46             return
47
48         name = name_entry.get()
49         symbol = symbol_entry.get()
50         supply = supply_entry.get()
51
52         if not name or not symbol or not supply:
53             messagebox.showwarning("Input Error", "All fields are required")
54             return
55
56         try:
57             supply = int(supply)
58
59             Token = web3.eth.contract(
60                 abi=contract_interface['abi'],
61                 bytecode=contract_interface['bin']
62             )
63
64             tx_hash = Token.constructor(name, symbol, supply).transact({
65                 'from': account
66             })
67
68             receipt = web3.eth.wait_for_transaction_receipt(tx_hash)
69
70             messagebox.showinfo(
71                 "Success",
72                 f"Token Deployed Successfully!\n\nContract Address:\n{receipt.contractAddress}"
73             )
74
75         except Exception as e:
76             messagebox.showerror("Error", str(e))
77
78         root = tk.Tk()
79         root.title("ERC20 Token Generator")
80         root.geometry("400x260")
81         root.resizable(False, False)
82
83         title = tk.Label(root, text="ERC20 Token Generator", font=("Arial", 14, "bold"))
84         title.pack(pady=15)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Traceback (most recent call last):
  File "c:/Users/parva/Downloads/Block chain\Ass 4.py", line 106, in <module>
    tk.Label(form, text="Total Supply").gr
AttributeError: 'Label' object has no attribute 'gr'
PS C:\Users\parva\Downloads\Block chain & C:/Users/parva/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/parva/Downloads/Block chain\Ass 4.py"
PS C:\Users\parva\Downloads\Block chain & C:/Users/parva/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/parva/Downloads/Block chain\Ass 4.py"
PS C:\Users\parva\Downloads\Block chain & C:/Users/parva/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/parva/Downloads/Block chain\Ass 4.py"
PS C:\Users\parva\Downloads\Block chain [
```

5 Air: Moderate Now

The screenshot shows the Visual Studio Code interface. The code editor displays the file `Ass 4.py`, which contains Python code for a Tkinter application to generate an ERC20 token. The terminal below shows a stack trace and command-line output indicating a `AttributeError` related to the `gr` variable.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Traceback (most recent call last):
  File "c:/Users/parva/Downloads/Block chain/Ass 4.py", line 106, in <module>
    tk.Label(form, text="Total Supply").gr
AttributeError: 'Label' object has no attribute 'gr'
PS C:\Users\parva\Downloads\Block chain & C:/Users/parva/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/parva/Downloads/Block chain/Ass 4.py"
PS C:\Users\parva\Downloads\Block chain & C:/Users/parva/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/parva/Downloads/Block chain/Ass 4.py"
PS C:\Users\parva\Downloads\Block chain []

```

The screenshot shows the Visual Studio Code interface again. The code editor displays the same `Ass 4.py` file. This time, the terminal shows the command `python Ass 4.py` being run successfully, and a modal window titled "ERC20 Token Generator" is displayed, prompting for Token Name, Token Symbol, and Total Supply.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Traceback (most recent call last):
  File "c:/Users/parva/Downloads/Block chain/Ass 4.py", line 106, in <module>
    tk.Label(form, text="Total Supply").gr
AttributeError: 'Label' object has no attribute 'gr'
PS C:\Users\parva\Downloads\Block chain & C:/Users/parva/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/parva/Downloads/Block chain/Ass 4.py"
PS C:\Users\parva\Downloads\Block chain & C:/Users/parva/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/parva/Downloads/Block chain/Ass 4.py"
PS C:\Users\parva\Downloads\Block chain & C:/Users/parva/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/parva/Downloads/Block chain/Ass 4.py"

```

## Observation:

- **State variables** = name, symbol, decimals, totalSupply, balanceOf
- **Constructor** = constructor(uint initialSupply)
- **Token transfer function** = transfer(address to, uint value)
- **Balance storage** = mapping(address => uint) balanceOf
- **Event used** = Transfer(address from, address to, uint value)