LYCEUM OF THE PHILIPPINES UNIVERSITY-LAGUNA
MIDTERM EXAMINATION – ITPEL2L

Thomas Joaquin T. Bernardo

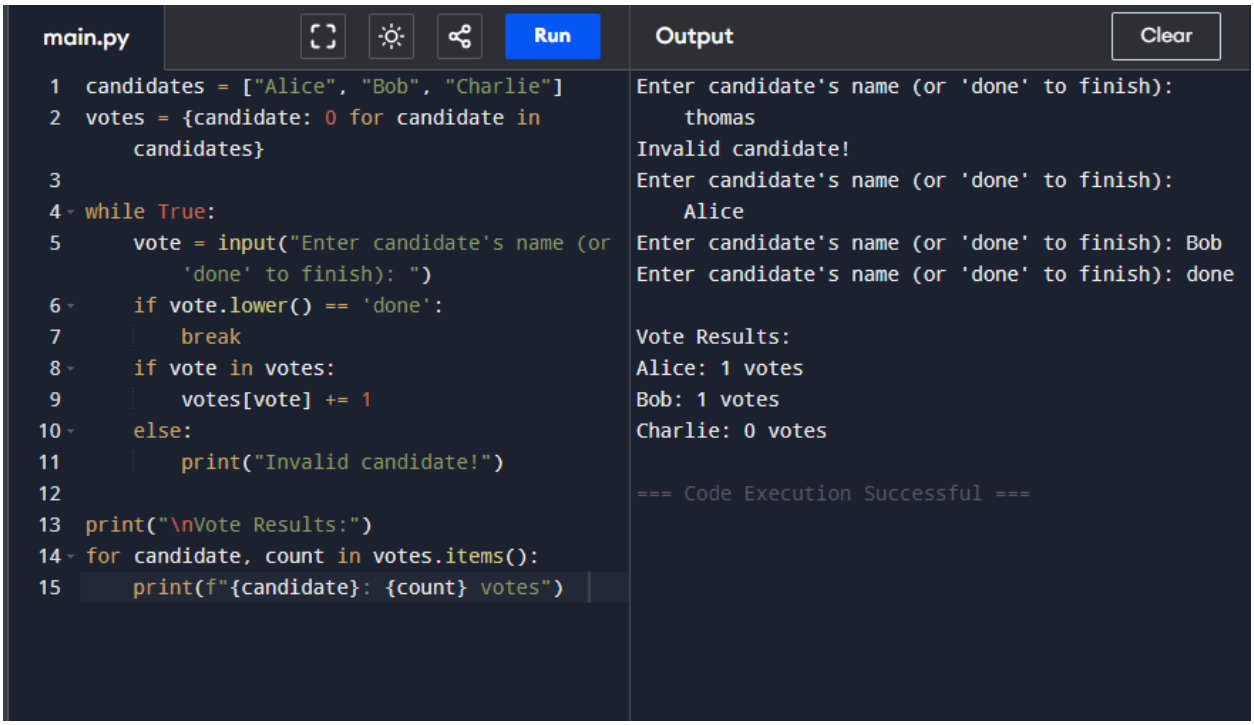
BSIT 2-1

PART I: PYTHON PROGRAMMING

1. **Output:** 24
    - Explanation: This is because the function calculates the factorial of 4 (4 × 3 × 2 × 1).
2. **Output:** [9, 1, 1, 25]
    - Explanation: the “squares” variable Squares odd numbers in the list [3, 1, 4, 1, 5].
3. **Output:** [3, 4, 5]
    - Explanation: Used for loop to add 2 to each element in the list [1, 2, 3].
4. **Output:** {'apple'}
    - Explanation: since it used and “&” symbol, the phyton knows this and Returns the intersection (common item) between sets x and y.
5. **Output:** evitcelE
    - Explanation: Reverses the string "Elective".

B. Problem Solving

Solution:



Solution:

```
candidates = ["Alice", "Bob", "Charlie"]
votes = {candidate: 0 for candidate in candidates}

while True:
    vote = input("Enter candidate's name (or 'done' to finish): ")
    if vote.lower() == 'done':
        break
    if vote in votes:
        votes[vote] += 1
    else:
        print("Invalid candidate!")

print("\nVote Results:")
for candidate, count in votes.items():
    print(f"{candidate}: {count} votes")
```

C. Code Completion

Solution:

```
def second_largest(numbers):
  unique_numbers = list(set(numbers))
  unique_numbers.sort()
  return unique_numbers[-2] if len(unique_numbers) >= 2 else None
```

PART II: INTRODUCTORY FLASK & HTML APPLICATION (50 POINTS)

A. Concept Questions (10 points)

1. This @app.route() will map a URL to a Python function.
2. render_template() and then renders an HTML template with dynamic content.
3. **Pass variables** using render_template('file.html', var_name=value).
4. **GET vs POST:**
    - GET: Data visible in URL (for queries).
    - POST: Data hidden (for sensitive input).
5. **Run Flask app:** Use flask run after setting FLASK_APP=filename.py.

B. Code writing

```
from flask import Flask, request, render_template_string

app = Flask(__name__)

@app.route('/')
def home():
  return "Welcome to MyBioApp!"

@app.route('/form', methods=['GET'])
def form():
  return '''
```

```
    <form action="/result" method="POST">
        Name: <input type="text" name="name"><br>
        Favorite Language: <input type="text" name="language"><br>
        <button type="submit">Submit</button>
    </form>
    '''

@app.route('/result', methods=['POST'])
def result():
    name = request.form['name']
    language = request.form['language']
    return f"Hello {name}! Your favorite language is {language}."

if __name__ == '__main__':
    app.run(debug=True)
```

C. Debugging and Improvement

Fixed Code:

```
@app.route('/greet')
def greet():
    name = request.args.get('name', 'Guest')  # Fixed: .get() method with default
    return f'Hello {name}'
```

D. Creative Task (

- **I will implement a homepage**

```
@app.route('/')
def home():
    return render_template('home.html', bio="I'm a Python developer!")
```

- **I will also add a contact form**

```
@app.route('/contact', methods=['GET', 'POST'])
def contact():
    if request.method == 'POST':
        email = request.form['email']
        return f"Thanks! We'll contact {email} soon."
    return render_template('contact.html')
```