Homework Assignment: Control Structures – Conditionals and Logical Operators

2.1 Conditional Statements

If Statement

Syntax:

```
if condition:
    # code block
```

Explanation:

- The `if` statement evaluates a condition. If the condition is `True`, the code block inside the `if` statement is executed.

Example:

```
age = 20

if age >= 18:
    print("You are an adult.")
```

Output:

```
You are an adult.
```

Elif and Else Statements

Syntax:

```
if condition1:
    # code block
```

```
elif condition2:
    # code block
else:
    # code block
```

Explanation:

- `elif` (short for "else if") allows you to check multiple conditions sequentially.
- `else` executes a block of code if none of the previous conditions are `True`.

Example:

```
age = 16

if age >= 18:
    print("You are an adult.")
elif age >= 13:
    print("You are a teenager.")
else:
    print("You are a child.")
```

Output:

```
You are a teenager.
```

Comparison Operators

- `==` : Equal to
- `!=` : Not equal to
- `>` : Greater than
- `<` : Less than
- `>=` : Greater than or equal to
- `<=` : Less than or equal to

Example:

```
temperature = 30

if temperature > 25:
```

```python
    print("It's hot outside.")
else:
    print("It's cool outside.")
```

Output:

```
It's hot outside.
```

2.2 Logical Operators

Logical operators allow you to combine multiple conditions in your conditional statements.

AND (and)

Explanation:

- Both conditions must be `True` for the entire expression to be `True`.

Example:

```python
age = 20
has_license = True

if age > 18 and has_license:
    print("Can drive")
else:
    print("Cannot drive")
```

Output:

```
Can drive
```

OR (or)

Explanation:

- At least one of the conditions must be `True` for the entire expression to be `True`.

Example:

```python
has_car = False
has_license = True

if has_car or has_license:
    print("Can travel")
else:
    print("Cannot travel")
```

Output:

```
Can travel
```

NOT(`not`)

Explanation:

- Inverts the boolean value of the condition.

Example:

```python
is_raining = False

if not is_raining:
    print("You don't need an umbrella.")
else:
    print("Take an umbrella.")
```

Output:

```
You don't need an umbrella.
```

Exercises

Complete each exercise by writing a separate Python file (`exercise1.py`, `exercise2.py`, etc.). Ensure your code runs without errors and produces the expected output.

Exercise 1: Check Voting Eligibility

Task:

Write a program that asks the user for their age and prints whether they are eligible to vote (18 years or older).

Example Output:

```
Enter your age: 20
You are eligible to vote.


Enter your age: 16
You are not eligible to vote.
```

Exercise 2: Determine the Largest Number

Task:

Write a program that takes three numbers as input and prints the largest one.

Example Output:

```
Enter first number: 10
Enter second number: 25
Enter third number: 15
The largest number is 25.


Enter first number: 5
Enter second number: 3
Enter third number: 8
The largest number is 8.
```

Exercise 3: Grade Classification

Task:

Create a program that takes a student's score (0-100) as input and prints the corresponding grade based on the following criteria:

- A: 90-100
- B: 80-89
- C: 70-79
- D: 60-69
- F: Below 60

Example Output:

```
Enter your score: 85
Your grade is: B


Enter your score: 72
Your grade is: C
```

Exercise 4: Password Validation

Task:

Write a program that asks the user to enter a password. The password must meet the following criteria:

- At least 8 characters long
- Contains both letters and numbers

Print "Password is valid." if it meets the criteria, otherwise print "Password is invalid."

Example Output:

```
Enter your password: pass1234
Password is valid.
```

```
Enter your password: pass
Password is invalid.
```

Exercise 5: Leap Year Checker

Task:

Write a program that takes a year as input and determines whether it is a leap year. A leap year is divisible by 4 but not by 100, unless it is also divisible by 400.

Example Output:

```
Enter a year: 2024
2024 is a leap year.
```

```
Enter a year: 1900
1900 is not a leap year.
```

Submission Guidelines

1. **Complete All Exercises:**

   Ensure each exercise is completed and saved in a separate Python file
   (`exercise1.py`, `exercise2.py`, etc.).

2. **Code Quality:**

   - Use meaningful variable names.
   - Include comments to explain your code where necessary.
   - Follow proper indentation and coding standards.

3. **Testing:**

   Run each script to verify that it works correctly and produces the expected output.

Good luck, and happy coding!