Qno.1 Explain the key features of Python that make it a popular choice for programming

Python is popular because it’s simple, readable, and flexible. Key features:
•Easy Syntax: Clear and simple code style.
•Interpreted: Runs line by line, helping with quick debugging.
•Libraries: Has many libraries for tasks like web dev, data science, and AI.
•Cross-Platform: Works on Windows, macOS, and Linux.
•Community: Large support network and good docs.
These make Python great for many uses.

Qno.2 Describe the role of predefined keywords in Python and provide examples of how they are used in a program

Predefined keywords in Python are reserved words that have special meanings and functions, defining the structure and behavior of code. They can’t be used as variable names. For example, if, for, and while control flow, def defines functions, and class creates classes.

- Example:

```
if x > 0:
   print("Positive")
else:
   print("Non-positive")
```

Qno.3 Compare and contrast mutable and immutable objects in Python with examples

In Python, mutable objects can be changed after creation, like lists and dictionaries. Immutable objects, like strings and tuples, cannot be altered once set. For example:

```
# Mutable
my_list = [1, 2, 3]
my_list[0] = 4  # Changes to [4, 2, 3]

# Immutable
my_string = "hello"
my_string[0] = "H"  # Raises an error
```

Qno.4 Discuss the different types of operators in Python and provide examples of how they are used

- Python has several types of operators:
    - 1. Arithmetic: Perform math operations (e.g., +, -, *, /).
- result = 5 + 3 # 8
    - 2. Comparison: Compare values (e.g., ==, !=, <, >).
- is_equal = 5 == 5 # True
    - 3. Logical: Combine conditions (e.g., and, or, not).
- valid = (5 > 3) and (2 < 4) # True
    - 4. Assignment: Assign values (e.g., =, +=, -=).
- x = 10
- x += 5 # 15
    - 5. Bitwise: Perform bit-level operations (e.g., &, |, ^).
- bitwise_result = 5 & 3 # 1
    - 6. Membership: Check membership (e.g., in, not in).
- is_in_list = 3 in [1, 2, 3] # True

Qno.5 Explain the concept of type casting in Python with examples

Type casting in Python is converting one data type to another, useful for handling different types in operations.

- Examples:
    - Integer to String:
- num = 5
- str_num = str(num) # "5"
    - String to Integer:

- age = "25"
- int_age = int(age) # 25

    - Float to Integer:

- pi = 3.14
- int_pi = int(pi) # 3

- Type casting allows flexibility with data manipulation.

Qno.6 How do conditional statements work in Python? Illustrate with examples.

Conditional statements in Python control program flow based on conditions using if, elif, and else. The code block runs if the condition is True; otherwise, it moves to the next condition or exits.

Example:

```
x = 10
if x > 5:
   print("Greater")
elif x == 5:
   print("Equal")
else:
   print("Smaller")
```

This checks x and prints accordingly.

Qno.7 Describe the different types of loops in Python and their use cases with examples.

Python has two main loop types: for loops and while loops.

•for loops iterate over a sequence (like a list, tuple, or string) and are useful for known, finite iterations.

```
for i in range(5):
   print(i)  # Prints 0 to 4
```

•while loops run as long as a condition is true, ideal for unknown or conditional iterations.

```
x = 0
while x < 5:
   print(x)
```

```
    x += 1  # Prints 0 to 4
```

These loops help automate repetitive tasks.