Assignment Python

Name: Anand Kr Gupta

Email ID: anandkg000@gmail.com

Phone N: 9026-477-478

Batch: Oct

Q1. Explain the key features of Python that make it a popular choice for programming.

Ans: Python is popular for several reasons:

- **Simple and Readable Syntax**: Python's syntax is designed to be easy to read and write, which makes it accessible for beginners and efficient for experienced developers.
- **Dynamic Typing**: Variables in Python do not require explicit declaration of their type, allowing for faster development and simpler code.
- **Interpreted Language**: Python code is executed line by line, which aids in debugging and allows for rapid testing of code.
- **Extensive Standard Library**: Python provides a vast library of built-in modules, making it easier to implement various functionalities without needing third-party libraries.
- **Large Community and Rich Ecosystem**: Python has a massive user community and a wealth of third-party libraries, making it suitable for a wide range of applications, from web development to data science.

Q2. Describe the role of predefined keywords in Python and provide examples of how they are used in a program.

Ans: Predefined keywords in Python serve as the foundational building blocks of the language. They are reserved words with specific meanings and cannot be used as variable names or identifiers. Keywords control the flow of the program, define variable scope, and help structure the code.

Examples of Keywords:

- if, elif, else: Used in conditional statements.

```
x = 10
if x > 5:
    print("x is greater than 5")
else:
    print("x is 5 or less")
```

- for, while: Used to define loops.

```
for i in range(3):
    print(i)
```

- def: Used to define a function.

```
def greet():
    print("Hello, world!")
greet()
```

Q3. Compare and contrast mutable and immutable objects in Python with examples.

Ans:

Mutable Objects: These are objects that can be modified after creation. Lists, dictionaries, and sets in Python are mutable.

```
# Modifies the first element
lst = [1, 2, 3]
lst[0] = 10
print(lst)
Output: [10, 2, 3]
```

Immutable Objects: These cannot be modified once they are created. Common immutable types are integers, floats, strings, and tuples.

```
# This will raise an error since strings are immutable

text = "hello"

text[0] = 'H'
```

Compare between Mutable and Immutable Objects in Python:

- **Mutability** is useful for collections where the contents are meant to change dynamically (e.g., lists).
- **Immutability** offers security by preventing accidental modifications (e.g., tuples used as dictionary keys).

Q4. Discuss the different types of operators in Python and provide examples of how they are used.

Ans: Python offers various operators for performing different operations on variables and values.

- **Arithmetic Operators**: Used for basic mathematical operations.

Examples: +, -, *, /, %, //, **

```
# Addition and division

a = 10

b = 3

print(a + b)

print(a / b)
```

- **Comparison Operators**: Used to compare two values.

Examples: ==, !=, >, <, >=, <=

```
#True or false

x = 5
```

```
print(x == 5)

print(x > 3)
```

- **Logical Operators**: Used to combine conditional statements.

Examples: and, or, not

```
# True or false

age = 20

print(age > 18 and age < 30)
```

- **Assignment Operators**: Used to assign values to variables.

Examples: =, +=, -=, *=

```
# Equivalent to x = x + 5

x = 10

x += 5

print(x)
```

Q5. Explain the concept of type casting in Python with examples.

Ans: Type casting converts one data type to another. Python supports both explicit and implicit type conversions.

- **Explicit Casting**: Using functions like int(), float(), str(), etc.

```
# Converts string "10" to integer 10

x = "10"

y = int(x)
```

- **Implicit Casting**: Automatically done by Python when performing operations involving different data types.

```
# x is implicitly converted to a float
x = 10
y = 3.5
z = x + y
```

Q6. How do conditional statements work in Python? Illustrate with examples.

Ans: Conditional statements are used to execute specific blocks of code based on conditions.

1. if Statement: Executes a block if the condition is true.
2. elif Statement: Used for multiple conditions.
3. else Statement: Executes if none of the conditions are true.

Example:

```
age = 18
if age < 18:
   print("Minor")
elif age == 18:
   print("Just an adult")
else:
   print("Adult")
```

Q7. Describe the different types of loops in Python and their use cases with examples.

Ans: Python provides two main types of loops- for and while.

1. **for Loop**: Iterates over a sequence (like a list or string).

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

2. **while Loop**: Repeats as long as a condition is true.

```
count = 0
while count < 5:
    print(count)
    count += 1
```

Use Cases:

- for loops are ideal for iterating over known ranges or collections.
- while loops are useful for scenarios where the end condition isn’t known in advance, such as reading lines from a file until the end is reached.

Each loop type provides break and continue statements for controlling loop flow:

- break exits the loop entirely.
- continue skips the current iteration and proceeds with the next one.