Name: Kandukuri Jennifer
Adm NO: 20126
Date of submission:

Holidays Assignment

SECTION - A

1. Explain programming and python in detail.

Definition and purpose :- programming is the process of creating a set of instructions that tell a computer how to perform a task. The purpose of programming is to automate tasks, solve problems, and create software applications that make our lives easier

Characteristics of python

* Code executes line by line
* Simple syntax similar to English
* Supports OOP concepts.
* Works on Windows, Mac, Linux

Applications

* Game development
* Data science and machine learning
* Web development
* Automation and scripting.

Types of comments in python

Single - line comment

This is a comment explaining the code

" " "

Multi-line comment (docstring)

Used for documentation

Can span multiple lines

" " ".

' ' '

Alternative multi-line comment

Using single quotes

' ' '

Importance of python in modern software development

* Strong community support

* Rapid development and prototyping

* Integration capabilities with other languages

* Industry standard for AI/ML and data science.

Describe data types and operators in python with suitable example.

1. Numeric Types:

```
# Integer
age = 30
print (type (age))  # < class '~~float~~'> 'int' >

# Float
price = 99.99
print (type (price))  # <class 'float'>
```

```
# complex
z = 3+4j
print (type (z)) # <class 'complex'>
```

2. Sequence Types:

```
# string
name = "Python"
print (type (name)) # <class 'str'>

# list (mutable)
fruits = ["apple", "banana", "cherry"]
print (type (fruits)) # <class 'list'>

# Tuple (immutable)
coordinates = (10, 20)
print (type (coordinates)) # <class 'tuple'>
```

3. Set Type

```
# set (unordered, unique elements)
unique_numbers = {1, 2, 3, 4}
print (unique_numbers) # {1, 2, 3, 4}
print (type (unique_numbers)) # <class 'set'>
```

4. Mapping Type:

```
# Dictionary (Key-value pairs)
student = {"name": "John", "age": 20, "grade": "A"}
print (type (student)) # <class 'dict'>
```

5. Boolean Type

```
is_active = True
is_valid = false
print (type (is_active)) # <class 'bool'>
```

Python operators

1. Arithmetic operators - These are used to perform mathematical calculations

operator	Meaning
+	Addition
-	subtraction
*	Multiplication
/	Division
%	Modulus (remainder)
//	floor division
**	Exponent (power)

```
    a = 10
    b = 3
print (a+b)
print (a%b)
print (a**b)
```

2. Assignment operator:- These are used to assign (or) update value in a variable

operator	example
=	x = 5
+=	x+ =3
-=	x-=2
*=	x*=2
/=	x/ =2

```
    x = 10
  x+ =5
print(x)
```

3. Comparison operators

* These are used to compare two values.
* Used to compare two values.

Operator	Meaning
==	Equal to
!=	Not equal
>	Greater than
<	Less than
>=	Greater than equal
<=	less than (or) equal

```
    a = 10
    b = 20
  print (a<b)
  print (a>b)
```

4. Logical operators

* Logical operators are used to combine multiple operations.
* They are commonly used in decision-making statements

Operator	Meaning
and	True if both are True
or	True if any one is True
not	Reverse the result.

```
    a = 10
    b = 20
  print(a<b & b>15)
```

5. Membership operator

* Membership operator checks whether a value exists in a sequence
* Used with lists, strings, tuples & sets.

operator	Meaning
in	present
no -1	not present

```
list 1 = [1, 2, 3]
print (2 in list 1)
print ('a' not in 'python')
```

6. Identity operator

* Identity operator check whether two variables refers to the same memory object.
* They do not compare values, but object identity.

operator	Meaning
is	same object
is not	different object

```
a = 10
b = 10
```

python Input and output (1/10)

3. Explain python Input and output operations in detail.

The Input () function:

The input () function is used to take user input. By default, it always returns a string data type.

```
# Basic Input
name = input ("Enter your name:")
print (" Hello, ", name)

# Default type is string
age = input (" Enter your age:")
print (type (age)) # <class 'str'>
```

Type conversion while taking input:

```
# converting to integer
age = int (input ("Enter your age:"))
print (type (age)) # <class 'int'>
```

```
# converting to float
price = float (input ("Enter price:"))
print (type (price)) # <class 'float'>
# converting to boolean (from string)
answer = input ("enter True or false:")
is_valid = answer.lower() == "true"
```

Taking multiple inputs:

```
# Method 1: Using split()
a,b = input ("Enter two numbers:").split()
print (f"a = {a}, b= {b}")
# Method 2: with type conversion
x,y = map (int, input (" Enter numbers:").split())
print (f" sum = {x+y}")
# Method 3: Taking list input
numbers = list (map(int, input (" Enter numbers:").split()))
print (f"Numbers: {numbers}")
```

The print() function and formatted output:

```
#Basic print
print ("Hello, world!")
#print multiple values
name = "Alice"
age = 25
print ("Name:", name, "Age:", age)
# Using seperator (sep)
print ("Python", "is", "awesome", sep="-")
# Output : python - is - awesome
# Using end parameter
print ("Hello", end=" ")
print (" world")
# Output : Hello world
# format specifiers - Method 1: f- strings
name  = "John"
score = 95.5
print (f" student: {name}, score: {score:2f}")
# format specifiers - Method 2: format()
print ("Name: {}, Age: { }".format ("Alice",25))
print ("price: {:.2f}", format (99.999)) # output:price:100.00
# format specifiers - Method 3: % operator
print ("Name: % s, Age : % d" % ("Bob", 30))
print ("Value: %.3f" % 3.14159) # Output: Value: 3.142
```

4. Discuss control statements and decision - Making statements in python

Control statements are programming constructs that controls the flow of execution in a program. They allow programmers to make decisions, repeat actions, and alter the sequential execution of code based on conditions.

Importance:

* Enable decision making in programs
* Allow repitition of code blocks
* Make programs dynamic & interactive
* Reduce code redundancy.

Types of control statements.

python control statements are broadly classified into three categories.

conditional / Decision - making: if, if-else, if-elif-else

Looping: for, while

Jump statements: break, continue pass.

1. Decision making statement: These are used to execute code based on conditions. They depend on boolean expressions (True/ falses

if statement

Executes a block of code only if the condition is True.

If the condition is false, the block is skipped.

Syntax:

```
if condition:
 statement
age = 20
if age>=18:
print ("Eligible to vote")
```

The message is printed only when age >=18 is True.

(2.) if - else statement

Executes one block if condition is True

Executes another block if condition is false.

Syntax

```
if condition:
 statements
else:
 statements
 num=5
if num% ==0
print ("even number")
else
print ("odd number")
```

(3) if - elif - else statement

Used to check multiple conditions

Elif means = "else if"

Nested if statements.

```
# Example : Login System
username = "admin"
password = "1234"
if  username == "admin":
  if  password == "1234":
   print ("Login succesful")
  else:
   print ("wrong password!")
  else:
   print ("User not found"!)
```

5. Write an essay on python programming fundamentals.

Introduction : python has emerged as one of the most popular programming languages in the world, known for its simplicity, readability and versatility. This essay explores the fundamental concepts of python programming including its role in problem - solving, syntax characteristics, documentation practices, data handling and control flow mechanisms.

Role of programming in problem solving

Programming is the art of translating human logic into machine - executable instructions. It serves as a bridge between human thought processes and computational power. In problem - solving, programming follows a systematic approach : understanding the problem, designing an algorithm, implementing the solution and testing for correctness. python with its intuitive syntax makes this process accessible to beginners while remaining powerful enough for complex applications in data science & web development.

Python Syntax Simplicity and Readability
python's design philosophy emphasizes code readability, guided by the principle that code is read more often that it is written " Unlike languages like C++ or Java, python used identifation to define code blocks instead of curly braces, enforcing clean and consistent formatting. This simplicity allows developers to focus on solving problems rather than fighting with syntax.

Use of comments for code documentation.
Comments are essential for code documentation and collaboration. python supports single-line comments using the hash symbol (#) and multi-line comments using triple quotes (''' or """). Docstrings, a special form of comments, provide documentation for functions, classes, & modules, accessible through python's built in help system. Good commenting practices tranform code from mere instructions into a readable narrative.

Data Type, Operators and Input/Output operations
python offers rich built-in data types including numeric types (int, float, complex) sequences (str, list, tuple) sets, dictionaries and booleans. This diversity allows programmers to choose appropriate data structures for their specific needs. python's operators - arthimetic, comparison, logical, assignment, membership and identity - provide comprehensive tools for data manipulation. These fundamental tools from the building blocks of any python program.

Control flow using decision making statements.
It gives programs the ability to respond differently based on conditions. python's if-elif-else construct allows programs to evaluate conditions and execute corresponding code blocks. Combined with comparison and logical operators, these control structures empower programmers to implement sophisticated decision trees and validation systems.

Conclusion.
Python's fundamentals - its readable syntax, comprehensive data types, versatile operators, intuitive I/O operations and powerful control structures - make it an ideal language for both learning programming and building professional applications. python's role in shaping the future of software development remains significant. Mastering these fundamentals provides a solid foundation for advancing into more complex programming concepts.

SECTION - B

Practical python programs.

1. Movie Ticket pricing

```
# Movie Ticket pricing system
age = int (input ("Enter age: "))
is 3D = int (input ("Is it a 3D movie? (1 for Yes, 0 for No): "))
# Determine based price based on age
if age < 13:
price = 150   # child
```

```
elif age > = 60:
   price = 200     # senior
else:
  price = 250     # Adult (13-59)
#Add 3D surcharge if applicable
if is3D == 1:
  price = price + 50
print (f"Final ticket price: ₹ {price}")
# Sample output:
# Enter age : 25
# Is it a 3D movie ? (1 for Yes, 0 for No): 1
# final ticket price : ₹ 300
```

2. College Attendance Rule

```
# college Attendance Rule
attendance = float (input ("Enter attendance percentage:"))
medical_cert = int (input ("Do you have medical certificate ?
   (1 for Yes, 0 for No):"))
# check eligibility
if attendance > = 75:
 print ("Allowed")
elif attendance >=60 and medical_cert == 1:
  print ("Allowed")
else:
  print ("Not allowed)
# sample output:
# Enter attendance percentage : 65
# Do you have medical certificate ? (1 for Yes, 0 for No):
# Allowed
```

3. E-Commerce Discount

```
# E-commerce Discount calculator
bill = float (input ("Enter bill amount:"))
isPrime = int (input ("Are you a Prime number? (1 for yes,0 for No):"))
#Calculate discount percentage based on bill amount
if bill >= 5000:
    discount = 20
elif bill >= 2000:
    discount = 10
else:
    discount = 0
# Add extra 5% discount for Prime members
if isPrime == 1:
    discount = discount + 5
# Calculate final amount
discount_amount = bill + discount / 100
final_amount = bill - discount_amount
print (f" Bill amount: ₹ {bill}")
print (f" Discount: {discount}% (₹ {discount_amount})")
print (f" final Amount: ₹ {final_amount}")
# Sample output:
# Enter bill amount: 6000
# Are you a Prime member? (1 for Yes, 0 for No): 1
# Bill Amount : ₹ 6000.0
# Discount : 25% (₹1500.0)
# final Amount: ₹4500.0
```

4. Smartphone Battery Warning

```
# Smartphone Battery Warning System
battery = int (input (" Enter battery percentage: "))
is Charging = int (input ("Is phone charging? (1 for Yes,0 for No):"))
# check if charging first (priority condition)
```

```
if   isCharging == 1:
 print ("charging")
elif battery <= 20:
 print ("Low Battery")
elif battery <= 80:
 print ("Normal")
else:
  print ("full")
# Sample output:
# Enter battery percentage :15
# Is phone charging ? (1 for Yes, 0 for No):0
# Low battery.
```

5. Driving License Check

```
# Driving License Eligibility check
age = int (input ("Enter age:"))
testPassed = int (input ("Did you pass the driving test?
             (1 for Yes, 0 for No):"))
# Seniors (60+) don't need test, others need age >=8 AND
 test passed
if age >= 60:
 print (" Eligible")
elif age >=18 and test Passed ==1:
 print (" Eligible")
else:
 print ("Not Eligible")
# Sample output:
# Enter age:20
# Did you pass the driving test? (1 for Yes, 0 for No): 1
# Eligible.
```

```
if   isCharging == 1:
 print ("charging")
elif battery <= 20:
 print ("Low Battery")
elif battery <= 80:
 print ("Normal")
else:
  print ("full")
# Sample output:
# Enter battery percentage :15
# Is phone charging ? (1 for Yes, 0 for No):0
# Low battery.
```

5. Driving License Check

```
# Driving License Eligibility check
age = int (input ("Enter age:"))
testPassed = int (input ("Did you pass the driving test?
              (1 for Yes, 0 for No):"))
# Seniors (60+) don't need test, others need age >= 8 AND
 test passed
if age >= 60:
 print ("Eligible")
elif age >= 18 and test Passed == 1:
 print ("Eligible")
else:
 print ("Not Eligible")
# Sample output:
# Enter age:20
# Did you pass the driving test?(1 for Yes,0 for No):1
# Eligible.
```

6. Online food Delivery

```
#Online food Delivery -free delivery check
amount = float (input ("Enter order amount:"))
isGold = int (input ("Are you a Gold member? (1 for Yes,0 for No):"))
distance = float (input ("Enter delivery distance (in km):"))

# Distance >10 km means delivery is never free
if  distance > 10:
    print ("Delivery is NOT Free (distance too far)")
elif amount >= 500 or isGold ==1:
   print ("Delivery is free!")
else:
   print ("Delivery is NOT FREE")
#Sample Output:
# Enter order amount:400
#Are you a Gold member? (1 for Yes, 0 for No):1
# Enter delivery distance (in km):5
# Delivery is free!
```

7. Bank Loan Approval

```
#Bank Loan Approval system
salary = float (input ("Enter monthly salary:"))
creditscore = int (input ("Enter credit score:"))

#Loan approved if: (salary >= 30000 AND Credit >= 700) OR
salary >= 50000
if salary >=50000:
 print ("Loan Approved")
elif salary >= 30000 and creditscore >= 700:
 print ("Loan Approved")
else:
 print ("Loan Rejected")
```

```
# Sample output:
# Enter monthly salary: 35000
# Enter credit score: 720
# Loan Approved.
```

8. Electricity Bill

```
# Electricity Bill Calculator (No loops)
units = int (input ("Enter units consumed: "))
# Slab-based calculation without loops
if units < = 100:
    bill = units * 2
elif units < = 200:
    bill = (100 * 2) + (units -100) * 3
else:
    bill = (100 * 2) + (100 * 3) + (units - 200) * 5
print (f* units consumed : {units}")
print (f* Total Bill : ₹ {bill}")
# Breakdown:
# First 100 units : ₹ 2/unit = ₹ 200
# Next 100 units (101-200) : ₹ 3/unit = ₹ 300
# Above 200 units : ₹ 5/unit
# Sample Output:
# Enter units consumed : 250
# Units consumed : 250
# Total Bill : ₹ 750
```

9. Student scholarship

```
# student scholarship Eligibility
marks = float (input ("enter marks: "))
income = float (input ("Enter family income: "))
singleParent = int (input ("Is single parent child? (1 for Yes, 0 for No): "))

# scholarship : marks >= 85 AND (income < 500000 OR SINGLE parent)
if marks >= 85:
  if single Parent == 1:
  print ("Eligible for Scholarship")
 elif income < 500000:
  print ("Eligible for Scholarship")
 else:
   print ("Not Eligible (income too high)")
else:
  print ("Not Eligible (marks too low)")
# Sample Output:
# Enter marks: 90
# Enter family income: 600000
# Is single parent child? (1 for Yes, 0 for No): 1
# Eligible for Scholarship.
```

10. Online Exam Result

```
# Online exam Result
theory = int (input ("Enter theory marks: "))
practical = int (input ("Enter practical marks: "))
total = theory + practical
# pass if : (theory >= 40 AND practical >= 40) OR total >= 100
if theory >= 40 and practical >= 40:
 print (f"Result: PASS (Total: {total})")
elif total >= 100:
 print (f" Result: PASS (Total: {total}) - Grace applied")
else:
    print (f"Result : FAIL (Total: {total})")
# Sample Output:
# Enter theory marks : 35
# Enter practical marks: 70
# Result : PASS (Total:105) - Grace applied.
```

11. Hotel Room Pricing

```
# Hotel Room Pricing
isWeekend = int (input ("Is it weekend? (1 for Yes, 0 for No): "))
daysStayed = int (input ("Enter number of days: "))
# Determine rate per day
if isWeekend == 1:
    rate = 4000
else:
    rate = 3000.
# calculate base bill
bill = rate * daysStayed
# Apply 15% discount for stays more than 3 days
if daysStayed > 3:
    discount = bill * 0.15
    bill = bill - discount
    print (f"Rate per day: ₹ {rate}")
    print (f"Days stayed : {daysStayed}")
    print (f"Discount (15%) : ₹ {discount}")
    print (f" Final Bill : ₹ {bill}")
else:
    print (f" Rate per day : ₹ {rate}")
    print (f" Days stayed: {days Stayed}")
    print (f"Final Bill : ₹ {bill}")
# Sample output:
# Is it weekend? (1 for Yes, 0 for No):1
# Enter number of days: 5
# Rate per day: ₹4000
# Days stayed:5
# Discount (15%): ₹ 3000.0
# final Bill: ₹ 17000.0
```

12. Gaming Level Unlock.

```
# Gaming level unlock system
score = int (input ("Enter score:"))
is Premium = int (input ("Do you have premium pass? (1 for Yes,
0 for No): "))
usedcheat = int (input ("Did you use cheat ? (1 for Yes, 0 for No):"))
# Cheating always denies access
if usedcheat == 1:
 print ("Access Denied - cheating detected!")
elif score >100 or is Premium == 1:
 print (" Level Unlocked!")
else:
 print (" Level locked - sore more or get Premium!")
# Sample output:
# Enter score: 80
# Do you have premium pass? (1 for Yes, 0 for No): 1
# Did you use cheat ? (1 for Yes, 0 for No):0
# Level Unlocked!
```

13. Mobile Data Usage

```
# Mobile Data Usage check
dataUsed = float (input ("Enter daily data used (in GB):"))
hasUnlimitedPlan = int ("input ("Do you have unlimited plan?
(1 for Yes, 0 for No): "))
isRoaming = int (input ("is roaming on? (1 for Yes, 0 for No): "))
# check conditions for unlimited data
if dataUsed <=2:
  print ("Unlimited Data Available")
elif hasUnlimited Plan == 1 and isRoaming == 0;
  print ("Unlimited Data Available")
elif hasUnlimited Plan == 1 and isRoaming == 1;
  print ("Data Limited - Unlimited plan doesn't work on
                                      roaming")
else:
  print ("Data limited - upgrade to unlimited plan")
# Sample output:
# Enter daily data used (in GB): 3
# Do you have unlimited plan? (1 for Yes, 0 for No): 1
# Is roaming on ? (1 for Yes, 0 for No):0
# Unlimited Data Available.
```

14. Office Entry System.

```
# Office Entry System
idValid = int (input ("Is ID card valid? (1 for Yes, 0 for No):"))
fingerprint = int (input ("Does fingerprint match? (1 for Yes, 0 for No):"))
facescan = int (input ("Does facescan match? (1 for Yes, 0 for No): "))
isHoliday = int (input ("Is today a holiday? (1 for Yes, 0 for No): "))
#Holiday denies entry for everyone
if isHoliday == 1:
  print ("Entry Denied - Office closed for holiday")
elif idvalid == 1 and (fingerprint ==1 or facescan ==1):
  print ("Entry Granted - Welcome!")
else:
  if idvalid ==0;
     print("Entry Denied - Invalid ID Card")
else:
  print ("Entry Denied - Biometric vertification failed")
# Sample output:
# Is ID card valid? (1 for Yes, 0 for No): 1
# Does fingerprint match? (1 for Yes, 0 for No): 0
# Does face scan match? (1 for Yes, 0 for No): 1
# Is today a holiday? (1 for Yes, 0 for No): 0
# Entry Granted - welcome!
```

15. Movie Rating Display

```
# Movie Rating Display system
averageRating = float(input("Enter average rating:"))
isEditorsChoice = int(input("Is it Editor's Choice? (1 for Yes, 0 for No):"))
# Editor's choice always shows "Recommended"
if isEditorsChoice == 1:
  print ("Recommended")
elif averageRating >= 8.5:
  print ("Excellent")
elif averageRating >= 6.0:
  print ("Good")
else:
  print (" Average")
# Sample output:
# Enter average rating: 7.5
# Is it Editor's Choice? (1 for Yes, 0 for No): 0
# Good.
```