K. S. INSTITUTE OF TECHNOLOGY

Bangalore – 560109

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CIE Question Paper Scrutiny format

Course Name	**Python Programming**
Course Code	**18AI52**
Course Incharge	**Dr. Vaneeta.M**
Academic year	**2022-2023**
Semester	**Vth**
CIE #	**IA - I**
Set	A☑ B ☐
Scrutiny parameters	
Whether questions are according to assessment plan?	Yes ☑ No☐ ; If No, Suggestions:
Whether questions prepared are within the covered syllabus?	Yes ☑ No ☐ ; If No, Suggestions:
Whether all questions are mapped to CO/PO properly?	Yes ☑ No ☐ ; If No, Suggestions:
Whether questions framed are according to Blooms level?	Yes ☑ No ☐ ; If No, Suggestions:
Whether marks distribution for each question are correct?	Yes ☑ No ☐ ; If No, Suggestions:
Whether questions paper follows the format displayed?	Yes ☑ No ☐ ; If No, Suggestions:
Difficulty level	Very High ☐ High ☐ Moderate ☑ Low☐
Percentage of Similarity questions in Set A & B	10%
Final decision	Accepted without corrections☑ Accepted with minor corrections☐ Not accepted☐

Mary

Signature with date
of CIE Question paper setter

Mary

Name and Signature with
date of CIE Question paper
Scrutiniser

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
FIRST INTERNAL TEST QUESTION PAPER 2022-23 ODD SEMESTER

USN | | | | | | | | | | |

SET-A

Degree : B.E
Branch : Artificial Intelligence & Machine Learning
Course Title : Python Programming
Date : 14-11-2022

Semester : V
Course Code:18AI52

Duration :90 Minutes
MaxMarks:30

Note: Answer **ONE full** question from each part.

K-Levels: K1-Remebering, K2-Understanding, K3-Applying, K4-Analyzing, K5-Evaluating, K6-Creating

Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	**Make use of** example and explain the concept of exception. Write a Python program with exception handling code to solve divide-by-zero error situation.	6	CO1	K3
(b)	**Identify** the use of input(), string concatenation and string replication using examples.	6	CO1	K3
(c)	**Identify** the usage of local and global variables as Local and Global Scope in Python programs and explain how can you force a variable in a function to refer to the global variable?	6	CO1	K3
	OR			
2(a)	**Utilize** a suitable example and explain the syntax of user defined functions, passing of parameters and return values from functions.	6	CO1	K3
(b)	**Make use of** a code snippet and explain the use of break and continue in python.	6	CO1	K3
(c)	**Develop** a Python function with suitable parameters to generate first N Fibonacci numbers. Display suitable error message if the condition for input value is not followed.	6	CO1	K3
	PART-B			
3(a)	**Identify** the output of following commands: a) L1=[1,2,3,4] L1.append([12,34]) b) S1=['A','a','Ab','cc'] S1.sort() c) bacon = ['Hello'] bacon.insert(1, 'world') d) L2=['a','b','c','z'] del L2[1] e) Name='America' 'za' in name f) Marks=(25,59,67) Marks[1]=45	6	CO2	K3
(b)	**Make use of** example and explain keys(),values(), get() and setdefault() dictionary methods in python.	6	CO2	K3
	OR			
4(a)	**Construct** the differences between tuple different and a list and explain with example the function used to convert list to tuple.	6	CO2	K3
(b)	**Develop** program that counts the number of occurrences of each letter in a string.	6	CO2	K3

Dr. Vaneeta. M
Name and Sign of Course In charge

Dr. Vaneeta. M
Name and Sign of Module Coordinator

HOD AIML

Principal

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

I SESSIONAL TEST QUESTION PAPER 2022-23 ODD SEMESTER

Set A

SCHEME AND SOLUTION

Degree	:	B.E	Semester	:	V
Branch	:	Artificial Intelligence & Machine Learning	Course Code	:	18AI52
Course Title	:	Python Programming	Max Marks	:	30

Q.NO.	POINTS	MARKS

1(a)	**Make use of** example and explain the concept of exception. Write a Python program with exception handling code to solve divide-by-zero error situation.
	Explanation 3M
	Program 3M
	• If we don't want to crash the program due to errors instead we want the program to detect errors, handle them, and then continue to run.
	• Errors can be handled with try and except statements.
	• The code that could potentially have an error is put in a try clause. The program execution moves to thestart of a following except clause if an error happens.
	`def spam(divideBy):`
	`  try:`
	`  return 42 / divideBy`
	`  except ZeroDivisionError:`
	`  print('Error: Invalid argument.')`
	`print(spam(2))`
	`print(spam(12))`
	`print(spam(0))`
	`print(spam(1))`
		3+3=6M

1(b)	Identify the use of input(), string concatenation and string replication using examples. Input(): 2M String Concatenation:2M String Replication : 2M	2+2+2=6M
1(c)	**Identify** the usage of local and global variables as Local and Global Scope in Python programs and explain how can you force a variable in a function to refer to the global variable? **Explanation:4M** **Example:2M**	4+2=6M

	• Parameters and variables that are assigned in a called function are said to exist in that function's *localscope.* • Variables that are assigned outside all functions are said to exist in the *global scope*. • A variable that exists in a local scope is called a *local variable*, while a variable that exists in the globalscope is called a *global variable.* • A variable must be one or the other; it cannot be both local and global. • If we need to modify a global variable and force a variable in a function to refer to the global variable from within a function, use the global statement.	
2(a)	**Utilize** a suitable example and explain the syntax of user defined functions, passing of parameters and return values from functions. Syntax of user defined function:2M Passing Parameters:2M Return Values:2M	2+2+2=6M
2 (b)	**Make use of** a code snippet and explain the use of break and continue in python. Explanation:4M Example:2M ***1. break Statements:*** ➢ There is a shortcut to getting the program execution to break out of a while loop's clause early. ➢ If the execution reaches a break statement, it immediately exits the while loop's clause. ***2. Continue statements:*** ➢ Like break statements, continue statements are used inside loops. ➢ When the program execution reaches a continue statement, the program execution immediately jumpsback to the start of the loop and reevaluates the loop's condition.	4+2=6M
2(c)	**Develop** a Python function with suitable parameters to generate first N Fibonacci numbers. Display suitable error message if the condition for input value is not followed. nterms = int(input("How many terms? ")) # first two terms n1, n2 = 0, 1 count = 0 # check if the number of terms is valid if nterms <= 0:	

```
  print("Please enter a positive integer")
# if there is only one term, return n1
elif nterms == 1:
  print("Fibonacci sequence upto",nterms,":")
  print(n1)
# generate fibonacci sequence
else:
  print("Fibonacci sequence:")
  while count < nterms:
    print(n1)
    nth = n1 + n2
    # update values
    n1 = n2
    n2 = nth
    count += 1
```

Q.NO.	PART B	
3 (a)	**Identify** the output of following commands: a) L1=[1,2,3,4,[12,34]] b) ['A', 'Ab', 'a', 'cc'] c) Helloworld d) ['a','c','z'] e) False f) Error	1 mark each=6M
3 (b)	**Make use of** example and explain keys(),values(), get() and setdefault() dictionary methods in python. keys()-1M values()-1M get()-2M setdefault()-2M	1+1+2+2=6M
4 (a)	**Construct** the differences between tuple different and a list and explain with example the function used to convert list to tuple. List is mutable; Elements can be modified. Tuple is immutable: Elements can not be modified. Explanation:4M Example:2M	4+2=6M

4 (b)	**Develop** program that counts the number of occurrences of each letter in a string.
	`message = 'It was a bright cold day in April, and the clocks were striking thirteen.'`
	`count = {}`
	
	`for character in message:`
	`  count.setdefault(character, 0)`
	`  count[character] = count[character] + 1`
	
	`print(count)`
		6M

K. S. INSTITUTE OF TECHNOLOGY

Bangalore – 560109

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CIE Question Paper Scrutiny format

Course Name	Python Programming
Course Code	18AI52
Course Incharge	Dr. Vaneeta.M
Academic year	2022-2023
Semester	Vth
CIE #	IA - I
Set	A☐ B☑
Scrutiny parameters	
Whether questions are according to assessment plan?	Yes☑ No☐ ; If No, Suggestions:
Whether questions prepared are within the covered syllabus?	Yes☑ No☐ ; If No, Suggestions:
Whether all questions are mapped to CO/PO properly?	Yes☑ No☐ ; If No, Suggestions:
Whether questions framed are according to Blooms level?	Yes☑ No☐ ; If No, Suggestions:
Whether marks distribution for each question are correct?	Yes☑ No☐ ; If No, Suggestions:
Whether questions paper follows the format displayed?	Yes☑ No☐ ; If No, Suggestions:
Difficulty level	Very High ☐ High ☐ Moderate ☑ Low☐
Percentage of Similarity questions in Set A & B	10%.
Final decision	Accepted without corrections☑ Accepted with minor corrections☐ Not accepted☐

Signature with date of CIE Question paper setter

Name and Signature with date of CIE Question paper Scrutiniser

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
FIRST INTERNAL TEST QUESTION PAPER 2022-23 ODD SEMESTER

USN | | | | | | | | | | |

SET-B

Degree : B.E **Semester : V**
Branch : Artificial Intelligence & Machine Learning **Course Code: 18AI52**
Course Title : Python Programming
Date : 14-11-2022

Duration :90 Minutes **MaxMarks:30**

Note: Answer **ONE full** question from each part.

K-Levels: K1-Remebering, K2-Understanding, K3-Applying, K4-Analyzing, K5-Evaluating, K6-Creating

Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	**Utilize** a code snippet and explain the use of break and continue in python.	6	CO1	K3
(b)	**Develop** a Python function with suitable parameters to generate Armstrong numbers between 1 to 1000	6	CO1	K3
(c)	**Make use of** example and explain following: a) range () b) random module c) key arguments in print()	6	CO1	K3
	OR			
2(a)	**Identify** the usage of local and global variables as Local and Global Scope in Python programs and explain how can you force a variable in a function to refer to the global variable?	6	CO1	K3
(b)	**Make use of** example and explain the concept of exception. Implement a code which prompts user for Celsius temperature, convert the temperature to Fahrenheit, and print out the converted temperature by handling the exception.	6	CO1	K3
(c)	**Utilize** an example and explain elif, for, while statements in Python.	6	CO1	K3
	PART-B			
3(a)	**Make use of** example and explain keys(),values(), get() and setdefault() dictionary methods in python.	6	CO2	K3
(b)	**Identify** the output of following commands: a) S1=['H','a','hb','CC'] S1.sort() b) C=[11,12,13,14] C.append([122,134]) c) Attendance=(25,59,67) Attendance[1]=45 d) S2 = ['How'] bacon.insert(1, 'are') S2.insert(2,'you') e) L2=['a','b','c','z'] del L2[3] f) Name='World is Big' 'or' in name	6	CO2	K3

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

I SESSIONAL TEST QUESTION PAPER 2022 – 23 ODD SEMESTER

Set B **SCHEME AND SOLUTION**

Degree	:	**B.E**	**Semester**	:	**V**
Branch	:	**Artificial Intelligence & Machine Learning**	**Course Code**	:	**18AI52**
Course Title	:	**Python Programming**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
1(a)	**Utilize** a code snippet and explain the use of break and continue in python. Explanation:4M Example:2M ***1. break Statements:*** ➢ There is a shortcut to getting the program execution to break out of a while loop's clause early. ➢ If the execution reaches a break statement, it immediately exits the while loop's clause. ***2. Continue statements:*** ➢ Like break statements, continue statements are used inside loops. ➢ When the program execution reaches a continue statement, the program execution immediately jumpsback to the start of the loop and reevaluates the loop's condition.	4+2=6M

1(b)	**Develop** a Python function with suitable parameters to generate Armstrong numbers between 1 to 1000.
	`def power(x, y):`
	`  if y == 0:`
	`  return 1`
	`  if y % 2 == 0:`
	`  return power(x, y // 2) * power(x, y // 2)`
	`  return x * power(x, y // 2) * power(x, y // 2)`
	`# Function to calculate order of the number`
	`def order(x):`
	`  # Variable to store of the number`
	`  n = 0`
	`  while (x != 0):`
	`  n = n + 1`
		6M

```
        x = x // 10

    return n
def isArmstrong(x):

    n = order(x)
    temp = x
    sum1 = 0

    while (temp != 0):
        r = temp % 10
        sum1 = sum1 + power(r, n)
        temp = temp // 10

    # If condition satisfies
    return (sum1 == x)

# Driver code
x = 153
print(isArmstrong(x))

x = 1253
print(isArmstrong(x))
```

1(c)	Make use of example and explain following: a) range () b) random module c) key arguments in print() range()-2M random module – 2M Key arguments(end and sep) 2M	2+2+2=6M
2(a)	**Identify** the usage of local and global variables as Local and Global Scope in Python programs and explain how can you force a variable in a function to refer to the global variable? **Explanation:4M** **Example:2M** • Parameters and variables that are assigned in a called function are said to exist in that function's *localscope.* • Variables that are assigned outside all functions are said to exist in the *global scope*. • A variable that exists in a local scope is called a *local variable*, while a variable that exists in the globalscope is called a *global variable.* • A variable must be one or the other; it cannot be both local and global. • If we need to modify a global variable and force a variable in a function to refer to the global variable from within a function, use the global statement.	4+2=6M
2 (b)	**Make use of** example and explain the concept of exception. Implement a code which prompts user for Celsius temperature, convert	4+2=6M

	the temperature to Fahrenheit, and print out the converted temperature by handling the exception. **Explanation 3M** **Program 3M** • If we don't want to crash the program due to errors instead we want the program to detect errors, handle them, and then continue to run. • Errors can be handled with try and except statements. • The code that could potentially have an error is put in a try clause. The program execution moves to thestart of a following except clause if an error happens.	
2(c)	Utilize an example and explain elif, for, while statements in Python. elif 2M for-2M while-2M	2+2+2=6M
Q.NO.	**PART B**	
3 (a)	**Make use of** example and explain keys(),values(), get() and setdefault() dictionary methods in python. keys()-1M values()-1M get()-2M setdefault()-2M	1+1+2+2=6M
3 (b)	a) ['CC', 'H', 'a', 'hb'] b) [11,12,13,14, [122,134]] c) Error d) Error e) ['a','b','c'] f) Error	1 Mark each=6M
4 (a)	Identify the output for following commands : IA = {'sub': 'math', 'marks': 42} 1) 'math', 42 2) 'sub', 'marks' 3) ('sub', 'math'), ('marks', 42) 4) Key: sub Value: math Key: marks Value: 42 5) {'sub': 'math', 'marks': 42,'semester': 2} 6) Error	1 Mark each=6M
4(b)	**Make use of** example and explain append (), insert() and remove() methods in list. **append()-2M, insert()-2M, remove()-2M**	2+2+2=6M

K. S. INSTITUTE OF TECHNOLOGY

Bangalore – 560109

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CIE Question Paper Scrutiny format

Course Name	**Python Programming**
Course Code	**18AI52**
Course Incharge	**Dr. Vaneeta.M**
Academic year	**2022-2023**
Semester	**Vth**
CIE #	**IA - II**
Set	A☑ B☐
Scrutiny parameters	
Whether questions are according to assessment plan?	Yes☑ No☐ ; If No, Suggestions:
Whether questions prepared are within the covered syllabus?	Yes☑ No☐ ; If No, Suggestions:
Whether all questions are mapped to CO/PO properly?	Yes☑ No☐ ; If No, Suggestions:
Whether questions framed are according to Blooms level?	Yes☑ No☐ ; If No, Suggestions:
Whether marks distribution for each question are correct?	Yes☑ No☐ ; If No, Suggestions:
Whether questions paper follows the format displayed?	Yes☑ No☐ ; If No, Suggestions:
Difficulty level	Very High☐ High☐ Moderate☑ Low☐
Percentage of Similarity questions in Set A & B	10%
Final decision	Accepted without corrections☑ Accepted with minor corrections☐ Not accepted☐

Mary

Signature with date of CIE Question paper setter

Mary

Name and Signature with date of CIE Question paper Scrutiniser

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU - 560109
SECOND INTERNAL TEST QUESTION PAPER 2022-23 ODD SEMESTER

SET: A

USN | | | | | | | | | | |

Degree	:	**B. E.,**	**Semester**	:	**V**
Branch	:	**Artificial Intelligence & Machine Learning**	**Course Code**	:	**18AI52**
Course Title	:	**Python Programming**	**Date**	:	**22/12/2022**
Duration	:	**90 Minutes**	**Max Marks**	:	**30**

Note: Answer **ONE full** question from each part.

K-Levels: K1-Remebering, K2-Understanding, K3-Applying, K4-Analyzing, K5-Evaluating, K6-Creating

Q. No.	Questions	Marks	CO	K-Level
	PART-A			
1(a)	**Explain** the following with example (i) Grouping with parenthesis (ii)Matching multiple groups with pipe (iii)Matching specific repetitions with braces	6	CO3	K2
(b)	**Develop** an example and define the following short hand characters (i){n,m} (ii)\w (iii)\s (iv)\d$ (v)\D (vi)[^abc]	6	CO3	K3
(c)	**Build** a python program for Phone number and Email address extractor by making use of Regular expression	6	CO3	K3
	OR			
2(a)	**Explain** the following with example: 1. Extract parts of file path 2. Opening a file 3. Read and write in file	6	CO3	K2
(b)	**Construct** the regular expression for following with an example: a. Matches a string that has a digit followed by zero or more b's. b. Construct the character class for sequences of one upper case letter followed by one or more lower case letters c. Matches a number at the beginning of string, word at the end of string, with optional punctuation. d. Numbers (0-9) of length between 1 to 3 in a given string.	6	CO3	K3
(c)	**Build** a program to search for lines that start with 'X' followed by any non-whitespace characters followed by ':' ending with number. Display the sum of all these number	6	CO3	K3
	PART –B			
3(a)	**Make use of** example and explain split, join, rjust, ljust, strip and center methods in python string.	6	CO2	K3
(b)	**Utilize** an example and explain class, attribute, pure function and modifiers.	6	CO4	K3
	OR			
4(a)	**Build** a program to accept a sentence and find the number of words, digits, uppercase and lowercase letters in sentence.	6	CO2	K3
(b)	**Demonstrate** pure functions and modifiers with examples.	6	CO4	K2

Vaneeta. M
Name & Signature of Course In charge

Vaneeta. M
Name & Signature of Module Coordinator

HOD AIML

Principal

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

II SESSIONAL TEST QUESTION PAPER 2022-23 ODD SEMESTER

Set A **SCHEME AND SOLUTION**

Degree	:	**B.E**	**Semester**	:	**V**
Branch	:	**Artificial Intelligence & Machine Learning**	**Course Code**	:	**18AI52**
Course Title	:	**Python Programming**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
1(a)	**Explain** the following with example (i) Grouping with parenthesis (ii)Matching multiple groups with pipe (iii)Matching specific repetitions with braces (i) Grouping with parenthesis (2M) Adding parentheses will create *groups* in the regex: (\d\d\d)-(\d\d\d-\d\d\d\d). Then you can use the group() match object method to grab the matching text from just one group. Example (ii)Matching multiple groups with pipe(2M) The | character is called a pipe. You can use it anywhere you want to match one of many expressions. For example, the regular expression r'Batman|Tina Fey' will match either 'Batman' or 'Tina Fey'. When both Batman and Tina Fey occur in the searched string, the first occurrence of matching text will be returned as the Match object. Example (iii)Matching specific repetitions with braces(2M) If you have a group that you want to repeat a specific number of times, follow the group in your regex with a number in braces. For example, the regex (Ha){3} will match the string 'HaHaHa', but it will not match 'HaHa', since the latter has only two repeats of the (Ha) group. Example	2+2+2=6M
1(b)	**Develop** an example and define the following short hand characters (i){n,m} (ii)\w (iii)\s (iv)\d$ (v)\D (vi)[^abc] 1 Mark each	1X6=6M
1(c)	**Build** a python program for Phone number and Email address extractor by making use of Regular expression import pyperclip, re phoneRegex = re.compile(r'''((\d{3}|\(\d{3}\))? # area code (\s|-|\.)? # separator (\d{3}) # first 3 digits	6M

```
(\s|-|\.) # separator
(\d{4}) # last 4 digits (\s*(ext|x|ext.)\s*(\d{2,5}))? # extension
)'", re.VERBOSE)
emailRegex = re.compile(r'"(
[a-zA-Z0-9._%+-]+ # username @ # @ symbol
[a-zA-Z0-9.-]+ # domain name (\.[a-zA-Z]{2,4}) # dot-something
)'", re.VERBOSE)
text = str(pyperclip.paste())
matches = []
for groups in phoneRegex.findall(text):
phoneNum = '-'.join([groups[1], groups[3], groups[5]]) if groups[8] != '':
phoneNum += ' x' + groups[8] matches.append(phoneNum)
for groups in emailRegex.findall(text):
matches.append(groups[0])
if len(matches) > 0: pyperclip.copy('\n'.join(matches)) print('Copied to
clipboard:') print('\n'.join(matches))
else:
print('No phone numbers or email addresses found.')
```

2(a) **Explain** the following with example: 2+2+1+1=6M
Extract parts of file path 2. Opening a file 3. Read and write in file

1. Extract parts of file path (2M)

The parts of a file path include the following:
The *anchor*, which is the root folder of the filesystem
On Windows, the *drive*, which is the single letter that often denotes a physical hard drive or other storage device
The *parent*, which is the folder that contains the file
The *name* of the file, made up of the *stem* (or *base name*) and the *suffix* (or *extension*)

2. Find contents and size of file(2M)
3. Opening a file (1M)
4. Read and write in file(1M)

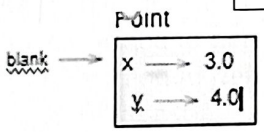
```
from pathlib import path
p=Path('spam.txt')
p.write_text("Hello World")
p.read_text()
h=spam.read()
print(h)
```

2 (b) **Construct** the regular expression for following with an example:**1.5 Marks each** 1.5x4=6M

a. Matches a string that has a digit followed by zero or more b's.
b. Construct the character class for sequences of one upper case letter followed by one or more lower case letters
c. Matches a number at the beginning of string, word at the end of string, with optional punctuation.

2(c) Numbers (0-9) of length between 1 to 3 in a given string.
Build a program to search for lines that start with 'X' followed by any non-whitespace characters followed by ':' ending with number. Display the sum of all these number

2(c)		6M
Q.NO.	**PART B**	
3 (a)	**Make use of** example and explain split, join, rjust, ljust, strip and center methods in python string. **1 Mark each** split: Method used to split the sentences based on separator. join: Method used to join words using a joining character ljust: Method to Left justify the string rjust: Method to right justify the string strip: to remove white spaces. Center: to align text to center	1 mark each=6M

3 (b)	Utilize an example and explain class, attribute, pure function and modifiers.
	□ Class: A programmer-defined type is also called a class. A class definition looks like this:
	`class Point:`
	`"""Represents a point in 2-D space."""`
	 
	
	The header indicates that the new class is called Point. The body is a docstring that ex-plains what the class is for. You can define variables and methods inside a class definition, but we will get back to that later.
	Attributes: assign values to an instance using dot notation. assigning values to named elements of an object. These elements are called attributes.
	Pure Functions: It does not modify any of the objects passed to it as arguments and it has no effect, like displaying a value or getting user input, other than returning a value.
	Modifiers: Sometimes it is useful for a function to modify the objects it gets as parameters. In that case, the changes are visible to the caller. Functions that work this way are called modifiers.
		1+1+2+2=6M

4 (a)	**Build** a program to accept a sentence and find the number of words, digits, uppercase and lowercase letters in sentence. str=input('Enter a string') W=str.split()	6M

4 (b)	UC,LC,DC=0,0,0
	for i in str:
	if i.isupper():
	UC+=1
	elif i.islower():
	LC+=1
	elif i.isdecimal():
	DC+=1
	print(" No. of words:", len(w) + "No of uppercase words:"+UC+"No. of lowercase words"+ LC+"No. of Digits"+DC)
	
	Demonstrate pure functions and modifiers with examples.
	
	Pure Functions: It does not modify any of the objects passed to it as arguments and it has no effect, like displaying a value or getting user input, other than returning a value.
	Modifiers: Sometimes it is useful for a function to modify the objects it gets as parameters. In that case, the changes are visible to the caller. Functions that work this way are called modifiers
		6M

Course incharge **Module Coordinator** **HOD**

K. S. INSTITUTE OF TECHNOLOGY

Bangalore – 560109

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CIE Question Paper Scrutiny format

Course Name	**Python Programming**
Course Code	**18AI52**
Course Incharge	**Dr. Vaneeta.M**
Academic year	**2022-2023**
Semester	**Vth**
CIE #	**IA - II**
Set	A☐ B☑
Scrutiny parameters	
Whether questions are according to assessment plan?	Yes ☑ No☐ ; If No, Suggestions:
Whether questions prepared are within the covered syllabus?	Yes ☑ No ☐ ; If No, Suggestions:
Whether all questions are mapped to CO/PO properly?	Yes ☑ No ☐ ; If No, Suggestions:
Whether questions framed are according to Blooms level?	Yes ☑ No ☐ ; If No, Suggestions:
Whether marks distribution for each question are correct?	Yes ☑ No ☐ ; If No, Suggestions:
Whether questions paper follows the format displayed?	Yes ☑ No ☐ ; If No, Suggestions:
Difficulty level	Very High ☐ High ☐ Moderate ☑ Low☐
Percentage of Similarity questions in Set A & B	10%
Final decision	Accepted without corrections☑ Accepted with minor corrections☐ Not accepted☐

Signature with date of CIE Question paper setter

Name and Signature with date of CIE Question paper Scrutiniser

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

II SESSIONAL TEST QUESTION PAPER 2022-23 ODD SEMESTER

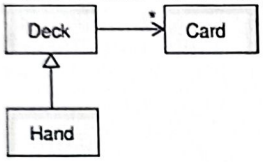
Set B — **SCHEME AND SOLUTION**

Degree	:	**B.E**	**Semester**	:	**V**
Branch	:	**Artificial Intelligence & Machine Learning**	**Course Code**	:	**18AI52**
Course Title	:	**Python Programming**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
1(a)	**Explain** the following with example (i) Grouping with parenthesis (ii)Matching multiple groups with pipe (iii)Matching specific repetitions with braces (i) Grouping with parenthesis (2M) Adding parentheses will create *groups* in the regex: (\d\d\d)-(\d\d\d-\d\d\d\d). Then you can use the group() match object method to grab the matching text from just one group. Example (ii)Matching multiple groups with pipe(2M) The | character is called a pipe. You can use it anywhere you want to match one of many expressions. For example, the regular expression r'Batman|Tina Fey' will match either 'Batman' or 'Tina Fey'. When both Batman and Tina Fey occur in the searched string, the first occurrence of matching text will be returned as the Match object. Example (iii)Matching specific repetitions with braces(2M) If you have a group that you want to repeat a specific number of times, follow the group in your regex with a number in braces. For example, the regex (Ha){3} will match the string 'HaHaHa', but it will not match 'HaHa', since the latter has only two repeats of the (Ha) group. Example	2+2+2=6M
1(b)	**Develop** an example and define the following short hand characters (i){n,m} (ii)\w (iii)\s (iv)\d$ (v)\D (vi)[^abc] 1 Mark each	1X6=6M
1(c)	Build a python program to find for lines having '@' sign between characters in a read text file.	6M

2(a)	**Explain** with example: 1. The Dollar and the Caret sign characters(3M) 2. Substituting the string with sub() method (3M) The caret symbol (^) at the start of a regex to indicate that a match must occur at the beginning of the searched text. Likewise, you can put a dollar sign ($) at the end of the regex to indicate the string must end with this regex pattern. And you can use the ^ and $ together to indicate that the entire string must match the regex that is, not enough for a match to be made on some subset of the string. **>>> beginsWithHello = re.compile(r'^Hello')** **>>> beginsWithHello.search('Hello, world!')** <re.Match object; span=(0, 5), match='Hello'> **>>> beginsWithHello.search('He said hello.') == None** True Regular expressions can not only find text patterns but can also substitute new text in place of those patterns. The sub() method for Regex objects is passed two arguments. The first argument is a string to replace any matches. The second is the string for the regular expression. The sub() method returns a string with the substitutions applied. **>>> namesRegex = re.compile(r'Agent \w+')** **>>> namesRegex.sub('CENSORED', 'Agent Alice gave the secret documents to Agent Bob.')** 'CENSORED gave the secret documents to CENSORED.'	3+3=6M
2 (b)	**Construct** the regular expression for following with an example:1.5 marks each a. Matches a string that has a **'A'** followed by zero or more B's b. Construct the character class for sequences of one-digit, upper case letter followed by one or more lower case letters c. Matches a number at the beginning of string, word at the end of string, with optional punctuation. Numbers (0-9) of length between 1 to 3 in a given string.	1.5x4=6M
2(c)	**Build** a python program for Phone number and Email address extractor by making use of Regular expression import pyperclip, re phoneRegex = re.compile(r'''((\d{3}|\(\d{3}\))? # area code (\s|-|\.)? # separator (\d{3}) # first 3 digits (\s|-|\.) # separator (\d{4}) # last 4 digits (\s*(ext|x|ext.)\s*(\d{2,5}))? # extension)''', re.VERBOSE) emailRegex = re.compile(r'''([a-zA-Z0-9._%+-]+ # username @ # @ symbol [a-zA-Z0-9.-]+ # domain name (\.[a-zA-Z]{2,4}) # dot-something)''', re.VERBOSE) text = str(pyperclip.paste()) matches = [] for groups in phoneRegex.findall(text): phoneNum = '-'.join([groups[1], groups[3], groups[5]]) if groups[8] != '': phoneNum += ' x' + groups[8] matches.append(phoneNum)	6M

	for groups in emailRegex.findall(text):
	matches.append(groups[0])
	if len(matches) > 0: pyperclip.copy('\n'.join(matches)) print('Copied to clipboard:')
	print('\n'.join(matches))
	else:
	print('No phone numbers or email addresses found.')	

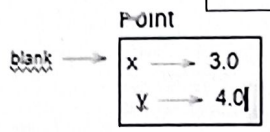
Q.NO.	**PART B**	
3 (a)	Utilize an example code and explain join(), sub() and split() string methods.**2 Mark each** split: Method used to split the sentences based on separator. join: Method used to join words using a joining character sub: Extract a substring	2 mark each=6M
3 (b)	Build an object diagram Construct a python program for a rectangle which as an attribute length and width and method area which returns the area of the rectangle. A class diagram is a more abstract representation of the structure of a program. Instead of showing individual objects, it shows classes and the relationships between them.  	6M

4 (a)	**Build** a program to accept a sentence from the user and display the longest word of that sentence along with its length.
	str=input('Enter a string')
	W=str.split()
	UC,LC,DC=0,0,0
	for i in str:
	if i.isupper():
	UC+=1
	elif i.islower():
	LC+=1
	elif i.isdecimal():
	DC+=1
	print(" No. of words:", len(w) + "No of uppercase words:"+UC+"No. of lowercase words"+ LC+"No. of Digits"+DC)
		6M

4 (b)	**Demonstrate** with an example class, attribute, pure function and modifiers.inition looks like this: Class: A programmer-defined type is also called a class. A class definition looks like this:	6M

```
class Point:

    """Represents a point in 2-D space."""
```



The header indicates that the new class is called Point. The body is a docstring that ex-plains what the class is for. You can define variables and methods inside a class definition, but we will get back to that later.

Attributes: assign values to an instance using dot notation. assigning values to named elements of an object. These elements are called attributes.

Pure Functions: It does not modify any of the objects passed to it as arguments and it has no effect, like displaying a value or getting user input, other than rēturning a value.

Modifiers: Sometimes it is useful for a function to modify the objects it gets as parameters. In that case, the changes are visible to the caller. Functions that work this way are called modifiers.

Course incharge **Module Coordinator** **HOD**

K. S. INSTITUTE OF TECHNOLOGY

Bangalore – 560109

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CIE Question Paper Scrutiny format

Course Name	**Python Programming**
Course Code	**18AI52**
Course Incharge	**Dr. Vaneeta.M**
Academic year	**2022-2023**
Semester	**Vth**
CIE #	**IA - III**
Set	A☑ B☐
Scrutiny parameters	
Whether questions are according to assessment plan?	Yes☑ No☐ ; If No, Suggestions:
Whether questions prepared are within the covered syllabus?	Yes☑ No☐ ; If No, Suggestions:
Whether all questions are mapped to CO/PO properly?	Yes☑ No☐ ; If No, Suggestions:
Whether questions framed are according to Blooms level?	Yes☑ No☐ ; If No, Suggestions:
Whether marks distribution for each question are correct?	Yes☑ No☐ ; If No, Suggestions:
Whether questions paper follows the format displayed?	Yes☑ No☐ ; If No, Suggestions:
Difficulty level	Very High☐ High☐ Moderate☑ Low☐
Percentage of Similarity questions in Set A & B	10%
Final decision	Accepted without corrections☑ Accepted with minor corrections☐ Not accepted☐

Mary

Signature with date of CIE Question paper setter

Mary

Name and Signature with date of CIE Question paper Scrutiniser

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
THIRD INTERNAL TEST QUESTION PAPER 2022-23 ODD SEMESTER

USN | | | | | | | | | | |

SET-A

Degree : B.E
Branch : Artificial Intelligence & Machine Learning
Course Title : Python Programming
Duration : 90 Minutes

Semester : V
Course Code: 18AI52
Date : 18-01-2023
Max Marks : 30

Note: Answer **ONE full** question from each part.
K-Levels: K1-Remembering, K2-Understanding, K3-Applying, K4-Analyzing, K5-Evaluating, K6-Creating

Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	**Demonstrate** how to download a file and save it to hard drive using request module, explain all the methods used in it.	6	CO5	K3
(b)	**Develop** a program that takes a number N from command line and creates an N x N multiplication table in excel spread sheet.	6	CO5	K3
(c)	**Explain** with and example encrypting text, decrypting and overlaying a watermark in PDFs files. Explain all the methods used.	6	CO5	K2
	OR			
2(a)	**Utilize** the getText() function for getting full text from a .docx file with example code and explain all the methods used	6	CO5	K3
(b)	**Develop** a program that find all the CSV files in the current working directory, read in the full contents of each file, write out the contents, skipping the first line, to a new CSV file.	6	CO5	K3
(c)	**Illustrate** the concept of parsing JSON using python code.	6	CO5	K2
	PART-B			
3(a)	**Illustrate** with python example the type-based dispatch	6	CO4	K2
(b)	**Develop** python code to overload "+" operator by providing the methods __add__ and explain the program methods and attributes	6	CO4	K3
	OR			
4(a)	**Develop** python code snippet and explain the usage of init and str methods.	6	CO4	K3
(b)	**Illustrate** the concepts of inheritance and class diagrams with examples.	6	CO4	K2

Dr. Vaneeta. M
Name and Sign of Course In charge

Dr. Vaneeta. M
Name and Sign of Module Coordinator

HOD AIML

Principal
Selected

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

III SESSIONAL TEST QUESTION PAPER 2022-23 ODD SEMESTER

Set A **SCHEME AND SOLUTION**

Degree	:	**B.E**	**Semester**	:	**V**
Branch	:	**Artificial Intelligence & Machine Learning**	**Course Code**	:	**18AI52**
Course Title	:	**Python Programming**	**Max Marks**	:	**30**

Q.NO.	POINTS	MARKS
1(a)	Demonstrate how to download a file and save it to hard drive using request module, explain all the methods used in it. import requests res = requests.get('http://www.gutenberg.org/cache/epub/1112/pg1112.txt') res.raise_for_status() playFile = open('RomeoAndJuliet.txt', 'wb') for chunk in res.iter_content(100000): playFile.write(chunk) Explanation of request, iter_content modules 3M Program 3M	3+3=6M
1(b)	Develop a program that takes a number N from command line and creates an N x N multiplication table in excel spread sheet. Program 6M	6M
1(c)	Explain with and example encrypting text, decrypting and overlaying a watermark in PDFs files. Explain all the methods used. Encrypting text: modules used encrypt(password) 2M Decrypting text: modules used decrypt(password) 2M Overlaying modules: 2M	2+2+2=6M

2(a)	Utilize the getText() function for getting full text from a .docx file with example code and explain all the methods used
	
	import docx
	def getText(filename):
	doc = docx.Document(filename) fullText = []
	for para in doc.paragraphs: fullText.append(para.text)
	return '\n'.join(fullText)
	The `getText()` function opens the Word document, loops over all the `Paragraph` objects in the `paragraphs` list, and then appends their text to the list in `fullText`. After the loop, the strings in `fullText` are joined together with newline characters. The *readDocx.py* program can be imported like any other module. Now if you just need the text from a Word document, you can enter the following:
	`>>> import readDocx`
	`>>> print(readDocx.getText('demo.docx'))`
		6M

2 (b)	Develop a program that find all the CSV files in the current working directory, read in the full contents of each file, write out the contents, skipping the first line, to a new CSV file.
	
	import csv, os
	os.makedirs ('headerRemoved', exist_ok=True)
	# Loop through every file in the current working directory. for csvFilename in os.listdir('.'):
	if not csvFilename.endswith('.csv'):
	continue # skip non-csv files
	print('Removing header from ' + csvFilename + '...')
	# Read the CSV file in (skipping first row). csvRows = []
	csvFileObj = open(csvFilename) readerObj = csv.reader(csvFileObj) for row in readerObj:
	if readerObj.line_num == 1: continue # skip first row
	csvRows.append(row) csvFileObj.close()
	# Write out the CSV file.
	csvFileObj = open(os.path.join('headerRemoved', csvFilename), 'w', newline='')
	csvWriter = csv.writer(csvFileObj) for row in csvRows:
	csvWriter.writerow(row) csvFileObj.close()
		6M

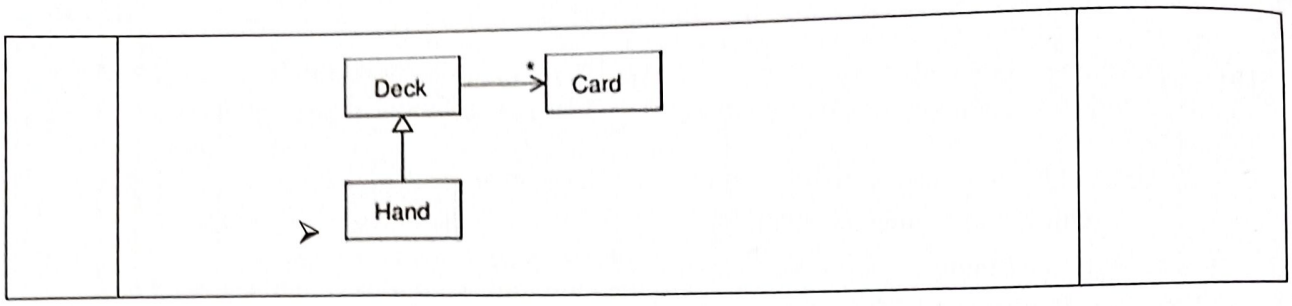
2(c)	Illustrate the concept of parsing JSON using python code.
	
	Reading JSON with the loads() Function 3M
	To translate a string containing JSON data into a Python value, pass it to the json.loads() function. (The name means "load string," not "loads.") Enter the following into the interactive shell:
	>>> stringOfJsonData = '{"name": "Zophie", "isCat": true, "miceCaught": 0, "felineIQ": null}'
	>>> import json
	>>> jsonDataAsPythonValue = json.loads(stringOfJsonData)
	>>> jsonDataAsPythonValue
	{'isCat': True, 'miceCaught': 0, 'name': 'Zophie', 'felineIQ': None}
	Writing JSON with the dumps() Function 3M
	The json.dumps() function (which means "dump string," not "dumps") will translate a Python value into a string of JSON-formatted data. Enter the fol- lowing into the interactive shell:
	>>> pythonValue = {'isCat': True, 'miceCaught': 0, 'name': 'Zophie', 'felineIQ': None}
	>>> import json
	>>> stringOfJsonData = json.dumps(pythonValue)
	>>> stringOfJsonData
	'{"isCat": true, "felineIQ": null, "miceCaught": 0, "name": "Zophie" }'
		3+3=6M
		
		6M

Q.NO.	**PART B**	

3 (a)	Illustrate with python example the type-based dispatch
	➢ type-based dispatch dispatches the computation to different methods based on the type of the arguments.
	➢ examples that use the + operator with different types:
	>>> start = Time(9, 45)
	>>> duration = Time(1, 35)
	>>> print(start + duration)11:20:00
	>>> print(start + 1337)
		6M

3 (b)	Develop python code to overload "+" operator by providing the methods __add__ and explain the program methods and attributes
	
	def _add(self, other): seconds=self.time_to_int()+other.time_to_int()
	return int_to_time(seconds)
	
	>>> start = Time(9, 45)
	>>> duration = Time(1, 35)
	>>> print(start + duration) 11:20:00
	
	➢ When + operator is applied to Time objects, Python invokes add .
	➢ Changing the behavior of an operator so that it works with programmer-defined types is called operator overloading.
	For every operator in Python there is a corresponding special method, like add
		6M

4 (a)	Develop python code snippet and explain the usage of init and str methods. **The init Method 3M** • The init method (short for "initialization") is a special method that gets invoked when an object is instantiated. • Its full name is init (two underscore characters, followed by init, and then two more underscores). • An init method for the Time class might look like this: # inside class Time: def _init (self, hour=0, minute=0, second=0): self.hour = hour self.minute = minute self.second = second **The _str_ Method 3M** • __str__is a special method, like init , that is supposed to return a string representation of an object. • # inside class Time: def__str_(self): return '%.2d:%.2d:%.2d' % (self.hour, self.minute, self.second)	3+3=6M
4 (b)	Illustrate the concepts of inheritance and class diagrams with examples. **Inheritance 3M** **Class Diagram 3M** A class diagram is a more abstract representation of the structure of a program. Instead of showing individual objects, it shows classes and the relationships between them.	3+3=6M



Mary
Course incharge

Mary
Module Coordinator

Mary
HOD

K. S. INSTITUTE OF TECHNOLOGY

Bangalore – 560109

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

CIE Question Paper Scrutiny format

Course Name	Python Programming
Course Code	18AI52
Course Incharge	Dr. Vaneeta.M
Academic year	2022-2023
Semester	Vth
CIE #	IA - III
Set	A☐ B☑
Scrutiny parameters	
Whether questions are according to assessment plan?	Yes☑ No☐ ; If No, Suggestions:
Whether questions prepared are within the covered syllabus?	Yes☑ No☐ ; If No, Suggestions:
Whether all questions are mapped to CO/PO properly?	Yes☑ No☐ ; If No, Suggestions:
Whether questions framed are according to Blooms level?	Yes☑ No☐ ; If No, Suggestions:
Whether marks distribution for each question are correct?	Yes☑ No☐ ; If No, Suggestions:
Whether questions paper follows the format displayed?	Yes☑ No☐ ; If No, Suggestions:
Difficulty level	Very High ☐ High ☐ Moderate ☑ Low☐
Percentage of Similarity questions in Set A & B	1 to 2%
Final decision	Accepted without corrections☑ Accepted with minor corrections☐ Not accepted☐

Mary

Signature with date
of CIE Question paper setter

Mary

Name and Signature with
date of CIE Question paper
Scrutiniser

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
THIRD INTERNAL TEST QUESTION PAPER 2022-23 ODD SEMESTER

USN | | | | | | | | | | |

SET-B

Degree : B.E
Branch : Artificial Intelligence & Machine Learning
Course Title : Python Programming
Duration : 90 Minutes

Semester : V
Course Code : 18AI52
Date : 18-01-2023
Max Marks : 30

Note: Answer **ONE full** question from each part.

K-Levels: K1-Remembering, K2-Understanding, K3-Applying, K4-Analyzing, K5-Evaluating, K6-Creating

Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	**Make use of** Beautiful Soup Module to parse HTML with the code snippet for creating finding an element and getting data explain the program.	6	CO5	K3
(b)	**Demonstrate** and explain the following with code snippet. i) Opening Excel documents with openpyxl. iii) Getting Cells, Rows and Columns from the Sheets.	6	CO5	K2
(c)	**Develop** a program to get a list of all files with the .pdf extension in the current working directory, sort them and write each page, excluding the first page, of each PDF to the output file.	6	CO5	K3
	OR			
2(a)	**Demonstrate** and explain the following with code snippet. i. Opening word documents. ii. Accessing paragraphs and runs from document. iii. Inserting header of different levels	6	CO5	K2
(b)	**Explain** the advantages of CSV files, Reader objects and Writer objects with python code.	6	CO5	K2
(c)	**Illustrate** the concept of parsing JSON using python code.	6	CO5	K2
	PART-B			
3(a)	**Utilize** an example python code and explain operator overloading in classes.	6	CO4	K3
(b)	**Make use of** code snippet and explain the concepts of inheritance and class diagrams.	6	CO4	K3
	OR			
4(a)	**Develop** python code snippet and explain the usage of init and str methods.	6	CO4	K3
(b)	**Illustrate** with python example the type-based dispatch.	6	CO4	K2

Dr. Vaneeta. M
Name and Sign of Course In charge

Dr. Vaneeta. M
Name and Sign of Module Coordinator

HOD AIML

Principal

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

III SESSIONAL TEST QUESTION PAPER 2022-23 ODD SEMESTER

Set B

SCHEME AND SOLUTION

Degree : B.E
Branch : Artificial Intelligence & Machine Learning
Course Title : Python Programming
Semester : V
Course Code : 18AI52
Max Marks : 30

Q.NO.	POINTS	MARKS

1(a)	Make use of Beautiful Soup Module to parse HTML with the code snippet for creating finding an element and getting data explain the program.
	Beautiful Soup is a module for extracting information from an HTML page and is much better for this purpose than regular expressions. The BeautifulSoup module's name is bs4.
	import requests, bs4
	>>> res = requests.get('http://nostarch.com')
	>>> res.raise_for_status()
	>>> noStarchSoup = bs4.BeautifulSoup(res.text)
	>>> type(noStarchSoup)
	retrieve a web page element from a BeautifulSoup object by calling the select()method and passing a string of a CSS *selector* for the element you are looking for. Selectors are like regular expressions: They specify a pattern to look for, in this case, in HTML pages instead of general text strings.
	import bs4
	>>> exampleFile = open('example.html')
	>>> exampleSoup = bs4.BeautifulSoup(exampleFile.read())
	>>> elems = exampleSoup.select('#author')
	>>> type(elems)
	<class 'list'>
		6M

1(b)	Demonstrate and explain the following with code snippet.
	i) Opening Excel documents with openpyxl.
	iii) Getting Cells, Rows and Columns from the Sheets.
	
	i) Opening Excel Documents with OpenPyXL 3M
	Once you've imported the openpyxl module, you'll be able to use the openpyxl
	.load_workbook() function. Enter the following into the interactive shell:
	>>> import openpyxl
	>>> wb = openpyxl.load_workbook('example.xlsx')
	>>> type(wb)
	<class 'openpyxl.workbook.workbook.Workbook'>
	The openpyxl.load_workbook() function takes in the filename and returns a value of the workbook data type. This Workbook object represents the Excel file, a bit like how a File object represents an opened text file.
		3+3=6M

	ii) Getting Cells from the Sheets 3M
	Once you have a Worksheet object, you can access a Cell object by its name.
	Enter the following into the interactive shell:
	>>> import openpyxl
	>>> wb = openpyxl.load_workbook('example.xlsx')
	>>> sheet = wb.get_sheet_by_name('Sheet1')
	>>> sheet['A1']
	<Cell Sheet1.A1>
	>>> sheet['A1'].value
	datetime.datetime(2015, 4, 5, 13, 34, 2)
	>>> c = sheet['B1']
	>>> c.value
	'Apples'
	>>> 'Row ' + str(c.row) + ', Column ' + c.column + ' is ' + c.value
	'Row 1, Column B is Apples'
	>>> 'Cell ' + c.coordinate + ' is ' + c.value
	'Cell B1 is Apples'
	>>> sheet['C1'].value
	73	

1(c)	Develop a program to get a list of all files with the .pdf extension in the current working directory, sort them and write each page, excluding the first page, of each PDF to the output file.
	
	import PyPDF2, os
	# Get all the PDF filenames. pdfFiles = []
	for filename in os.listdir('.'):
	if filename.endswith('.pdf'):
	pdfFiles.append(filename)
	pdfFiles.sort(key/str.lower)
	pdfWriter = PyPDF2.PdfFileWriter()
	# Loop through all the PDF files. for filename in pdfFiles:
	pdfFileObj = open(filename, 'rb')
	pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
	# Loop through all the pages (except the first) and add them.
	for pageNum in range(1, pdfReader.numPages):
	pageObj = pdfReader.getPage(pageNum)
	pdfWriter.addPage(pageObj)
	# Save the resulting PDF to a file.
	pdfOutput = open('allminutes.pdf', 'wb')
	pdfWriter.write(pdfOutput)
	pdfOutput.close()
		6M

2(a)	Demonstrate and explain the following with code snippet.
	i. Opening word documents.
	ii. Accessing paragraphs and runs from document.
	iii. Inserting header of different levels
	>>> import docx
	>>> doc = docx.Document('demo.docx')
	>>> len(doc.paragraphs)
	7
	>>> doc.paragraphs[0].text
		2+2+2=6M

'Document Title'
>>> len(doc.paragraphs[1].runs)
4
>>> doc.paragraphs[1].runs[0].text
'A plain paragraph with some '
>>> doc.paragraphs[1].runs[1].text
'bold'
>>> doc.paragraphs[1].runs[2].text
' and some '
>>> doc.paragraphs[1].runs[3].text
'italic'

Adding Headings

Calling add_heading() adds a paragraph with one of the heading styles. Enter the following into the interactive shell:

```
>>> doc = docx.Document()
>>> doc.add_heading('Header 0', 0)
<docx.text.Paragraph object at 0x00000000036CB3C8>
>>> doc.add_heading('Header 1', 1)
<docx.text.Paragraph object at 0x00000000036CB630>
```

2 (b) Explain the advantages of CSV files, Reader objects and Writer objects with python code. 2+2+2=6M

CSV files are simple, lacking many of the features of an Excel spread- sheet. For example, CSV files 2M

- Don't have types for their values—everything is a string
- Don't have settings for font size or colour
- Don't have multiple worksheets
- Can't specify cell widths and heights
- Can't have merged cells
- Can't have images or charts embedded in them

Reader Objects 2M

To read data from a CSV file with the `csv` module, you need to create a `Reader` object. A `Reader` object lets you iterate over lines in the CSV file. Enter the following into the interactive shell, with *example.csv* in the current working directory:

```
>>> import csv
>>> exampleFile = open('example.csv')
>>> exampleReader = csv.reader(exampleFile)
>>> exampleData = list(exampleReader)
>>> exampleData
```

Writer Objects 2M

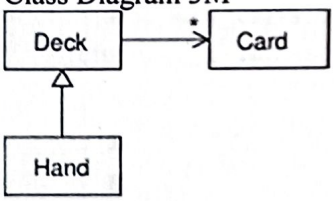
A `Writer` object lets you write data to a CSV file. To create a `Writer` object, you use the `csv.writer()` function. Enter the following into the interactive shell:

```
>>> import csv
>>> outputFile = open('output.csv', 'w', newline='')
>>> outputWriter = csv.writer(outputFile)
>>> outputWriter.writerow(['spam', 'eggs', 'bacon', 'ham'])
21
```

2(c)	Illustrate the concept of parsing JSON using python code.
	Reading JSON with the loads() Function 3M
	To translate a string containing JSON data into a Python value, pass it to the json.loads() function. (The name means "load string," not "loads.") Enter the following into the interactive shell:
	>>> stringOfJsonData = '{"name": "Zophie", "isCat": true, "miceCaught": 0, "felineIQ": null}'
	>>> import json
	>>> jsonDataAsPythonValue = json.loads(stringOfJsonData)
	>>> jsonDataAsPythonValue
	{'isCat': True, 'miceCaught': 0, 'name': 'Zophie', 'felineIQ': None}
	Writing JSON with the dumps() Function 3M
	The json.dumps() function (which means "dump string," not "dumps") will translate a Python value into a string of JSON-formatted data. Enter the fol- lowing into the interactive shell:
	>>> pythonValue = {'isCat': True, 'miceCaught': 0, 'name': 'Zophie', 'felineIQ': None}
	>>> import json
	>>> stringOfJsonData = json.dumps(pythonValue)
	>>> stringOfJsonData
	'{"isCat": true, "felineIQ": null, "miceCaught": 0, "name": "Zophie" }'	

Q.NO.	**PART B**	

3 (a)	Utilize an example python code and explain operator overloading in classes.
	def _sub(self, other): seconds=self.time_to_int()-other.time_to_int()
	return int_to_time(seconds)
	
	>>> start = Time(9, 45)
	>>> duration = Time(1, 35)
	>>> print(start - duration) 8:10:00
	➢ When + operator is applied to Time objects, Python invokes sub .
	➢ Changing the behavior of an operator so that it works with programmer-defined types is called operator overloading.
	For every operator in Python there is a corresponding special method, like sub.
		6M

3 (b)	Make use of code snippet and explain the concepts of inheritance and class diagrams. Inheritance 3M Class Diagram 3M  A class diagram is a more abstract representation of the structure of a program. Instead of showing individual objects, it shows classes and the relationships between them.	3+3=6M
4 (a)	Illustrate with python example the type-based dispatch ➢ type-based dispatch dispatches the computation to different methods based on the type of the arguments.	6M

	➢ examples that use the + operator with different types:
	>>> start = Time(9, 45)
	>>> duration = Time(1, 35)
	>>> print(start + duration)11:20:00
	>>> print(start + 1337)
	10:07:17	

4 (b)	Develop python code to overload "+" operator by providing the methods __add__ and explain the program methods and attributes
	def _add(self, other): seconds=self.time_to_int()+other.time_to_int()
	return int_to_time(seconds)
	
	>>> start = Time(9, 45)
	>>> duration = Time(1, 35)
	>>> print(start + duration) 11:20:00
	➢ When + operator is applied to Time objects, Python invokes add .
	➢ Changing the behavior of an operator so that it works with programmer-defined types is called operator overloading.
	For every operator in Python there is a corresponding special method, like add
		6M

Course incharge **Module Coordinator** **HOD**