INTERNSHIP REPORT
AT HDLC TECHNOLOGIES,

Attended by

S.Siva Vardhan Reddy (18121097)

in partial fulfilment for the award of the degree of

Bachelor of Technology

In

ELECTRONICS AND COMMUNICATION

from 02.09.2020 to 06.10.2020

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL SCIENCES

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

PADUR 603 103

HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)
CHENNAI

BONAFIDE CERTIFICATE

This is to certify that the "Internship report" submitted by S.Siva Vardhan Reddy is the work done by him and submitted during 2020–2021 academic year, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in ELECTRONICS AND COMMUNICATION , at HDLC TECHNOLOGIES.

HEAD OF THE DEPARTMENT
Dr.A.L.Vallikannu

INTERNSHIP COORDINATOR
Ms.K.Thenkumari

CERTIFICATION

HDLC TECHNOLOGIES

CERTIFICATE No. : HDLC/IT/PY/1459

08/10/2020

INTERNSHIP CERTIFICATE

This is to certify that **SETTIPALLI SIVAVARDHAN REDDY**, studying Electronics and Communication Engineering in Hindustan Institute of Technology and Science has successfully completed Internship program on **Python Programming** between 02/09/2020 & 06/10/2020 in our organization. During the time of Internship he has worked on various modules of python and successfully completed the project. We wish him all the very best for his future endeavors.

For HDLC Technologies Ltd

Authorized signatory

HDLC Technologies
No. 66, Pavithra Square, Chennai-600061
Email: hdlctech@gmail.com , cto@hdlctech.in

TABLE OF CONTENT

1.Introduction

HDLC is a startup company based in Chennai, India working in IT Industry with a focus on building software products using diffeent AI Technologies like Machine Learning, Deep Learning, Natural Language Processing, Speech recognition in specific segments in which we have expertise and business relationships. The startup is boot-strapped and self-funded**.**

We also offer a wide range of internship programs in Computer science aimed at technical and non-technical concepts made with the aim of creating capable, creative and tech-savvy professionals who combine theoretical know-how with practical expertise to solve complex situations by finding foolproof solution.

HDLC Technologies provide an digital learning platform offering students best learning experience and help them for doing internships and projects.

In this Internship, I had learnt about Programming with python. Such that 35 Days of internship has completed.

2. AGENDA

Day 1 - Introduction to python and software requirements to download.
Day 2 - Python comments, variables and syntax
Day 3&4 - Python Data Types, Numbers and Casting
Day 5&6 - Python Strings, Booleans and Operators
Day 7&8 - Lists and Tuples
Day 9 - Sets and Dictionaries
Day 10 - Python If….els and While loop statements
Day 11 - Python For loop Statements
Day 12&13 - Python functions
Day 14 - Python Arrays
Day 15 - Python Classes and objects
Day 16 - Python Modules
Day 17&18 - Python String Formatting
Day 19&20 - Python Oops Concepts

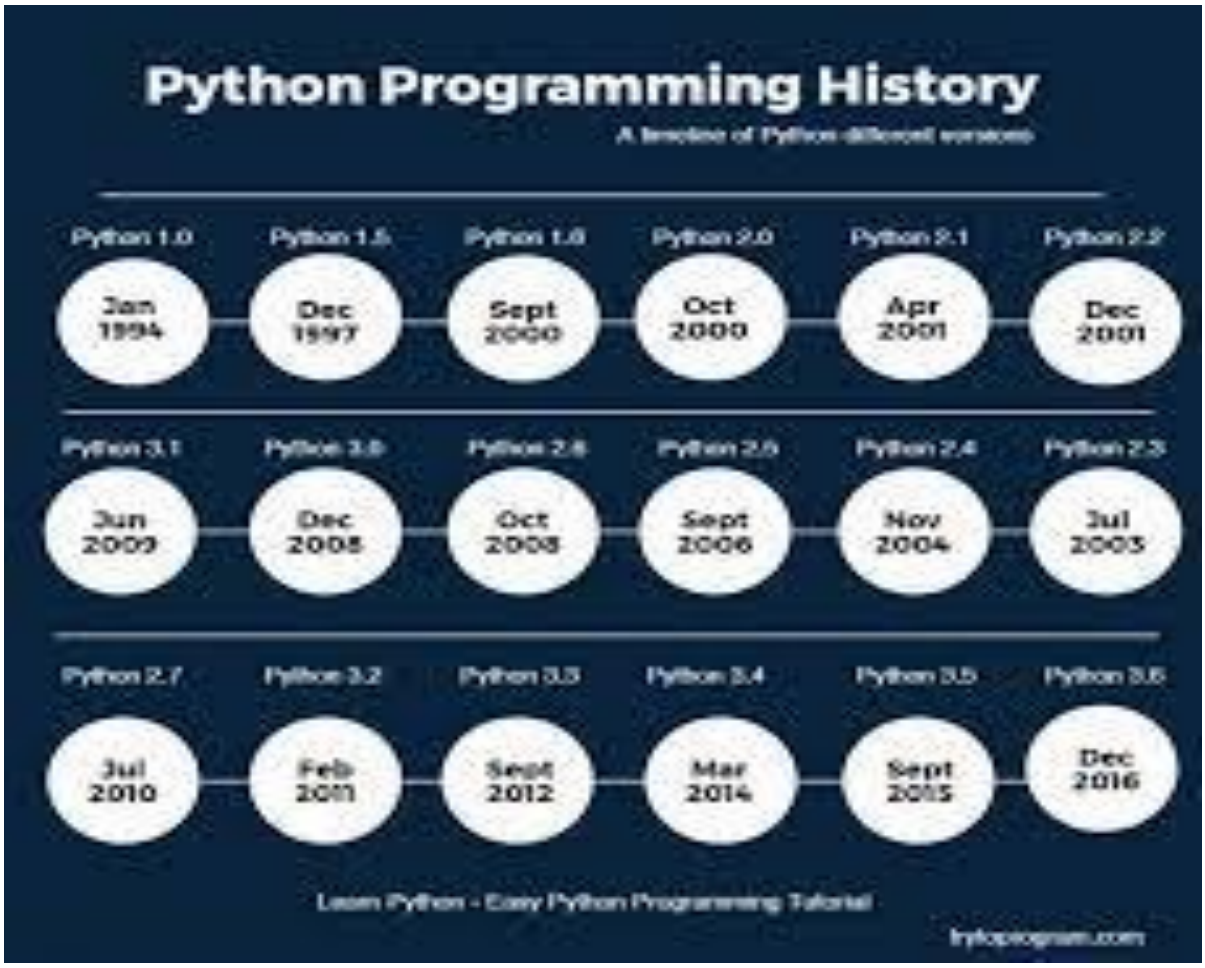
Day 21 & 22 - Python File handling
Day 23 - Python User Input
Day 24 - Break, Continue and Pass Statements
Day 25 - Module Quiz
Day 26 - Swap two variables
Day 27&28 - Printing patterns
Day 29 - Programming Questions from Level 0
Day 30 - Programming Questions from Level 1
Day 31 - Programming Questions from Level 2
Day 32 - Programming Questions from Level 3

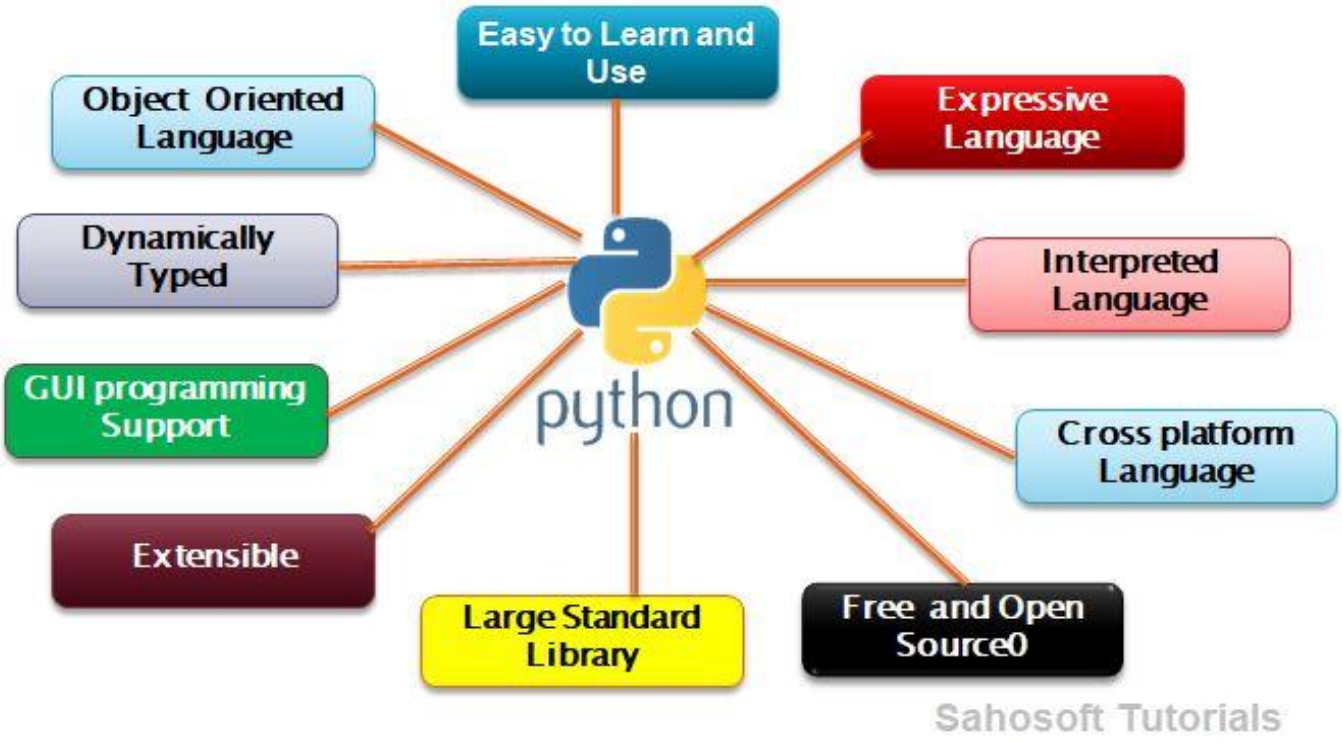
3. Topics

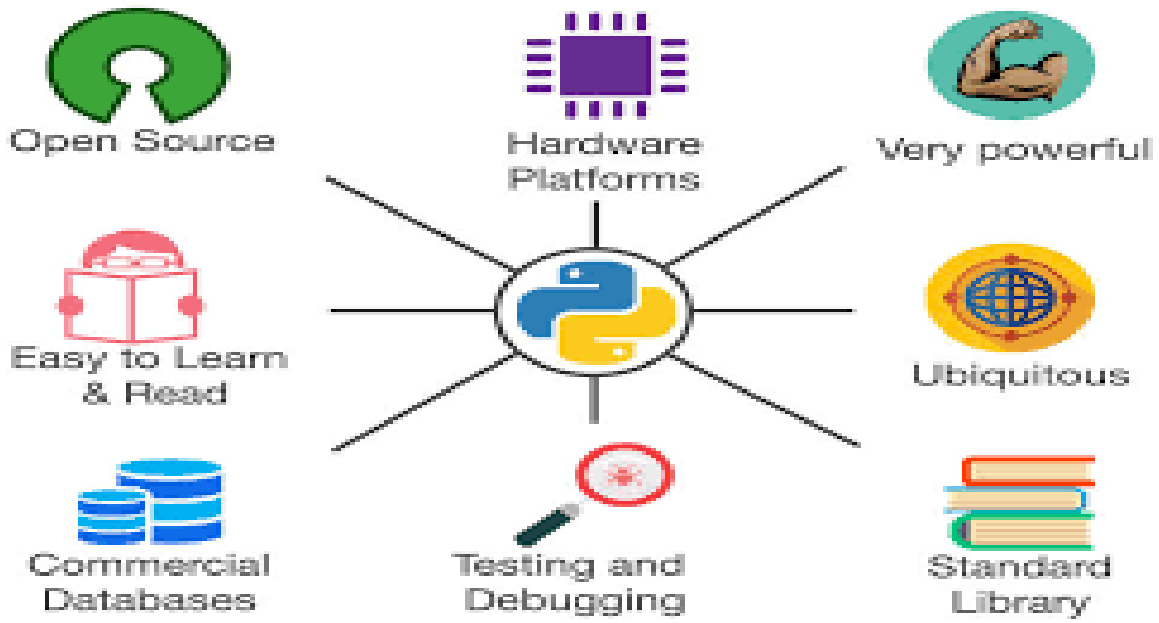
3.1 Python History

Python is a widely used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. The two of the most used versions have to do with Python 2.x & 3.x. There is a lot of competition between the two and both of them seem to have quite a number of different fanbase.

For various purposes such as developing, scripting, generation and software testing, this language is utilised. Due to its elegance and simplicity, top technology organisations like Dropbox, Google, Quora, Mozilla, Hewlett-Packard, Qualcomm, IBM, and Cisco have implemented Python.




Easy to Learn and Use
Object Oriented Language
Expressive Language
Dynamically Typed
Interpreted Language
python
GUI programming Support
Cross platform Language
Extensible
Large Standard Library
Free and Open Source0
Sahosoft Tutorials


Open Source
Hardware Platforms
Very powerful
Easy to Learn & Read
Ubiquitous
Commercial Databases
Testing and Debugging
Standard Library

3.2 Features in Python

There are many features in Python, some of which are discussed below –

1. Easy to code:

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer-friendly language.

2. Free and Open Source:

Python language is freely available at the official website and you can download it from the given download link below click on the **Download Python** keyword.

Download Python

Since it is open-source, this means that source code is also available to the public. So you can download it, use it as well as share it.

3. Object-Oriented Language:

One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, object encapsulation, etc.

4. GUI Programming Support:

Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python.

PyQt5 is the most popular option for creating graphical apps with Python.

5. High-Level Language:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6. Extensible feature:

Python is a Extensible language. We can write some Python code into C or C++ language and also we can compile that code in C/C++ language.

7. Python is Portable language:

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

8. Python is Integrated language:

Python is also an Integrated language because we can easily integrate python with other languages like c, c++, etc.

9. Interpreted Language:

Python is an Interpreted Language because Python code is executed line by line at a time. Unlike other languages C, C++, Java, etc. there is no need to compile python code; this makes it easier to debug our code. The source code of python is converted into an immediate form called **bytecode**.

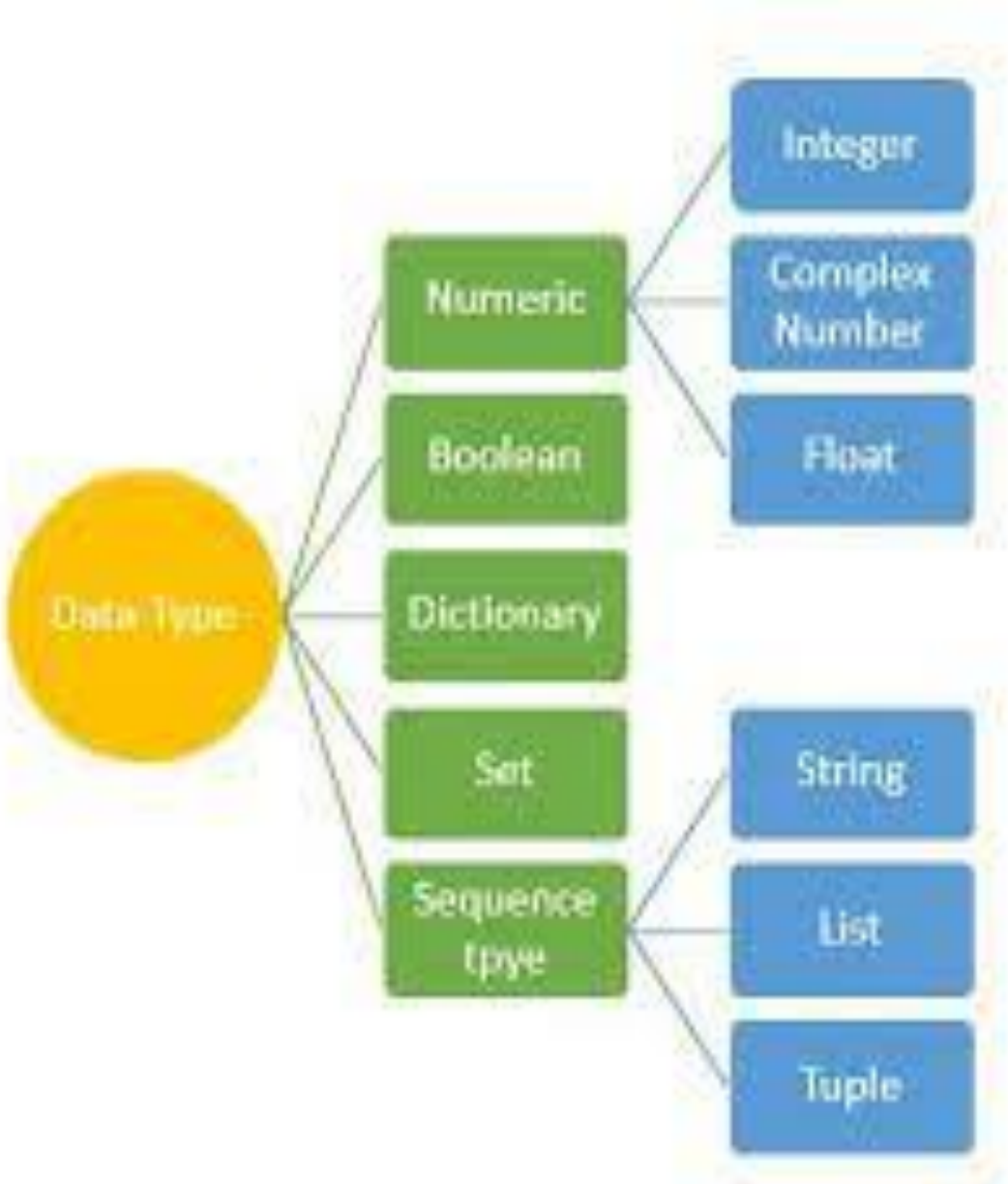
10. Large Standard Library

Python has a large standard library which provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in python such as regular expressions, unit-testing, web browsers, etc.

11. Dynamically Typed Language:

Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

3.3 Data Types in Python



3.3.1 Numeric

In Python, numeric data types represent the data which has numeric value. Numeric values can be integers, floating numbers or even complex numbers. These values are defined as int, float and complex classes in Python.

- **Integers** – This value is represented by int class. It contains positive or negative whole numbers (without fraction or decimal). In Python there is no limit to how long an integer value can be.
- **Float** – This value is represented by the float class. It is a real number with floating point representation. It is specified by a decimal point. Optionally, the character e or E followed by a positive or negative integer may be appended to specify scientific notation.
- **Complex Numbers** – Complex numbers are represented by complex classes. It is specified as (real part) + (imaginary part)j. For example – 2+3j.

3.3.2 Sequence Type

In Python, sequence is the ordered collection of similar or different data types. Sequences allow you to store multiple values in an organised and efficient fashion. There are several sequence types in Python –

- String
- List
- Tuple

1) String

In Python, Strings are arrays of bytes representing Unicode characters. A string is a collection of one or more characters put in a single quote, double-

quote or triple quote. In python there is no character data type, a character is a string of length one. It is represented by the str class.

Creating String

Strings in Python can be created using single quotes or double quotes or even triple quotes.

2) List

Lists are just like the arrays, declared in other languages which is an ordered collection of data. It is very flexible as the items in a list do not need to be of the same type.

Creating List

Lists in Python can be created by just placing the sequence inside the square brackets[].

3) Tuple

Just like list, tuple is also an ordered collection of Python objects. The only difference between tuple and list is that tuples are immutable i.e. tuples cannot be modified after it is created. It is represented by tuple class.

Creating Tuple

In Python, tuples are created by placing a sequence of values separated by 'comma' with or without the use of parentheses for grouping of the data sequence. Tuples can contain any number of elements and of any datatype (like strings, integers, list, etc.).

3.3.3 Boolean

Data type with one of the two built-in values, True or False. Boolean objects that are equal to True are truthy (true), and those equal to False are falsy (false). But non-Boolean objects can be evaluated in Boolean context as well and determined to be true or false. It is denoted by the class bool.

Note – True and False with capital ‘T’ and ‘F’ are valid booleans otherwise python will throw an error.

3.3.4 Set

In Python, Set is an unordered collection of data types that is iterable, mutable and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

Creating Sets

Sets can be created by using the built-in set() function with an iterable object or a sequence by placing the sequence inside curly braces, separated by ‘comma’. Type of elements in a set need not be the same, various mixed-up data type values can also be passed to the set.

3.3.5 Dictionary

Dictionary in Python is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only a single value as an element, Dictionary holds key:value pair. Key-value is provided in the dictionary to make it more optimised. Each key-value pair in a Dictionary is separated by a colon :, whereas each key is separated by a ‘comma’.

Creating Dictionary

In Python, a Dictionary can be created by placing a sequence of elements within curly {} braces, separated by ‘comma’. Values in a dictionary can be of any datatype and can be duplicated, whereas keys can’t be repeated and must be immutable. Dictionary can also be created by the built-in function

dict(). An empty dictionary can be created by just placing it in curly braces{}.

4. Assessment

4.1 Assessment

Assessment is done for the Programming Questions and quiz.

Programming questions:

a. **Check if a number is positive or negative in python.**
b. **Greatest of the Three numbers**
c. **Sum of N natural numbers**
d. **Reverse of a number**
e. **Palindrome number**
f. **Sum of elements in an array**

(a)
```
num = int(input())
 if num ==0:
      print('Zero')
elif num>0:
      print('Positive')
else:
      print('Negative')
```

```
num = int(input())
if num ==0:
  print('Zero')
elif num>0:
  print('Positive')
else:
  print('Negative')
```

```
15
Positive
```

(b) num1, num2, num3 = 10 , 30 , 20

```
max = 0
if num1 >= num2 and num1 >= num3:
    print(num1)
elif num2 >= num1 and num2 >= num3:
    print(num2)
else:
    print(num3)
```

```
num1, num2, num3 = 10 , 30 , 20
max = 0
if num1 >= num2 and num1 >= num3:
  print(num1)
elif num2 >= num1 and num2 >= num3:
  print(num2)
else:
  print(num3)
```

```
30
```

© number = 6
sum = int((number * (number + 1))/2)
print(sum)

```
number = int(input())
sum = int((number * (number + 1))/2)
print(sum)
```

```
6
21
```

(d)

```
num = 1234
temp = num
reverse = 0
while num > 0:
    remainder = num % 10
    reverse = (reverse * 10) + remainder
    num = num // 10

print(reverse)
```

```
num = 1234
temp = num
reverse = 0
while num > 0:
    remainder = num % 10
    reverse = (reverse * 10) + remainder
    num = num // 10

print(reverse)
```

```
4321
```

(e)

```
num = 1221
temp = num
reverse = 0
while temp > 0:
    remainder = temp % 10
    reverse = (reverse * 10) + remainder
    temp = temp // 10
if num == reverse:
  print('Palindrome')
else:
  print("Not Palindrome")
```

```
num = 1221
temp = num
reverse = 0
while temp > 0:
    remainder = temp % 10
    reverse = (reverse * 10) + remainder
    temp = temp // 10
if num == reverse:
  print('Palindrome')
else:
  print("Not Palindrome")
```

```
Palindrome
```

(e)

```
arr = [10, 89, 9, 56, 4, 80, 8]
Sum = 0

for i in range(len(arr)):
   Sum = Sum + arr[i]
print (Sum)
```

```
arr = [10, 89, 9, 56, 4, 80, 8]
Sum = 0

for i in range(len(arr)):
   Sum = Sum + arr[i]
print (Sum)
```

```
256
```

4.2 Viva Voce

1. What is the maximum length of a Python identifier?

- [ ] 32
- [ ] 16
- [ ] 128
- [ ] No fixed length is specified.

Show Explanation

2. What will be the output of the following code snippet?

```
print(2**3 + (5 + 6)**(1 + 1))
```

- [ ] 129
- [ ] 8
- [ ] 121
- [ ] None of the above.

Show Explanation

3. What will be the datatype of the var in the below code snippet?

```
var = 10
print(type(var))
var = "Hello"
print(type(var))
```

- [ ] str and int
- [ ] int and int
- [ ] str and str
- [ ] int and str

Show Explanation

4. How is a code block indicated in Python?

- [ ] Brackets.
- [ ] Indentation.
- [ ] Key.
- [ ] None of the above.

Show Explanation