

Section-3

d1
Ques.

Polymorphism :-

In programming language and type theory, polymorphism is the provision of single interface to entities of different types or the use of a single symbol to represent multiple different types.

The most commonly recognized major classes of polymorphism are :-

(i) Ad hoc polymorphism.

(ii) Parametric Polymorphism.

(iii) Subtyping Polymorphism.

Dynamic (Run time) Polymorphism :-

It is defined as the polymorphism existed at run-time. Here, Java compiler does not understand which method is called at compilation time. Only JVM decides which method is called at "run-time."

- Method overloading & methods are the examples for dynamic polymorphism.

For example :- Consider an application that serializes and de-serializes different types of documents.

- Static (Compile time) Polymorphism :-

It is defined as the polymorphism exhibited at compile time.

- Here, Java Compiler knows which method is called.

- Method overloading and method overriding using static methods; method overriding using private or final methods are examples for static polymorphism.

For example :- An employee object may have two print () methods one taking no arguments and one taking no arguments and one taking prefix string to be displayed along with the employee data.

*> Unpacking Sequence in Python :-

Python allows unpacking of any sequence (iterable) into variable using a simple assignment operation.

- Unpacking can be done by assigning sequence (iterable) to comma separated variables.
- Let us take example of unpacking sequence :-

Example :- In the below example of tuple P are assigned to variable X & Y.

```
# unpacking P into variable X and Y
P = (100, 200)
Print (X)
Print (Y)
```

Output :-

```
100
200
```

- In the below example list language is unpacked into english :-

```
# unpack the list into variables
language =
["hello", "namaste", "Bonjour", "Konnichiwa"]
```

english, hindi, french, japanese = language

print ("english = { }, hindi = { }, french = { },
Japanese = { }.

format (english, hindi, french, japanese)

Output :-

english = hello, hindi = namaste, japanese = konnichiwa
french = Bonjour.

Tuple in Python :-

• A tuple is a collection of objects which ordered and immutable.

• Tuples are sequences, just like lists. The difference b/w tuples & lists are, the tuples cannot be changed unlike & tuples use parentheses, whereas lists use square brackets.

• Creating a tuple is as simple as putting different comma-separated values.

- Optionally you can put these comma-separated values b/w parentheses also.
- For example :-

```
 tup 1 = ('physics', 'chemistry', 1)  
 tup 2 = (1, 2, 3, 4, 5);  
 tup 3 = "a", "b", "c", "d";
```

The empty tuple is written as two parentheses containing nothing :-

```
 tup 1 = ();
```

↑ To write a tuple containing a single value you have to include a comma, even though there is only one value -

```
 tup = (50,);
```