

CHAPTER-2**DATA ABSTRACTION**

Choose the Best Answer (1 Mark)

- 1. Which of the following functions that build the abstract data type?**
a) Constructors b) Destructors c) recursive d) Nested **Ans. (a)**
- 2. Which of the following functions that retrieve information from the data type?**
a) Constructors b) Selectors c) recursive d) Nested **Ans. (b)**
- 3. The data structure which is a mutable ordered sequence of elements is called**
a) Built in b) List c) Tuple d) Derived data **Ans. (b)**
- 4. A sequence of immutable objects is called**
a) Built in b) List c) Tuple d) Derived data **Ans. (c)**
- 5. The data type whose representation is known are called**
a) Built in datatype b) Derived datatype c) Concrete datatype
d) Abstract datatype **Ans. (c)**
- 6. The data type whose representation is unknown are called**
a) Built in datatype b) Derived datatype
c) Concrete datatype d) Abstract datatype **Ans. (d)**
- 7. Which of the following is a compound structure?**
a) Pair b) Triplet c) single d) quadrat **Ans. (a)**
- 8. Bundling two values together into one can be considered as**
a) Pair b) Triplet c) single d) quadrat **Ans. (a)**
- 9. Which of the following allow to name the various parts of a multi-item object?**
a) Tuples b) Lists c) Classes d) quadrats **Ans. (c)**
- 10. Which of the following is constructed by placing expressions within square brackets?**
a) Tuples b) Lists c) Classes d) quadrats **Ans. (b)**

PART II

Answer the following questions (2 Marks)

1. What is abstract data type?

Abstract Data type is a type or class for objects whose behavior is defined by a set of value and a set of operations.

2. Differentiate constructors and selectors.

Constructors	Selectors
Constructors are functions that build the abstract data type.	Selectors are functions that retrieve information from the data type.

3. What is a Pair? Give an example.

Python provides a compound structure called Pair which is made up of list or Tuple.

Example for List is [10, 20].

Example for tuple: colour= ('red', 'blue', 'Green')

4. What is a List? Give an example.

- ❖ List is constructed by placing expressions within square brackets separated by commas. Such an expression is called a list literal.
- ❖ List can store multiple values.
- ❖ Each value can be of any type and can even be another list.

Example for List is [10, 20].

5. What is a Tuple? Give an example.

A tuple is a comma-separated sequence of values surrounded with parentheses. Tuple is similar to a list.

Example: colour= ('red', 'blue', 'Green')

PART III

Answer the following questions (3 Marks)

1. Differentiate Concrete data type and abstract datatype.

Abstract Data Type	Concrete Data Type
ADT only mentions what operations are to be performed but not how these operations will be implemented.	In concrete data representation, a definition for each function is known.
ADT does not specify how data will be organized in memory and what algorithms will be used for implementing the operations	Concrete data types or structures are direct implementations of a relatively simple concept.
ADT offer a high level view of a concept independent of its implementation.	

2. Which strategy is used for program designing? Define that Strategy.

‘Wishful Thinking’ strategy is used for program designing.

Definition:

Wishful Thinking is the formation of beliefs and making decisions according to what might be pleasing to imagine instead of by appealing to reality.

3. Identify which of the following are constructors and selectors?

- a) N1=number() - Constructor
- b) acceptnum(n1) - Selector
- c) displaynum(n1) - Selector
- d) eval(a/b) - Selector
- e) x,y= makeslope (m), makeslope(n) - Constructor
- f) display() - Selector

4. What are the different ways to access the elements of a list? Give example.

The elements of a list can be accessed in two ways.

The first way is via our familiar method of multiple assignment, which unpacks a list into its elements and binds each element to a different name.

```
lst := [10, 20]
```

```
x, y := lst
```

In the above example x will become 10 and y will become 20.

A second method for accessing the elements in a list is by the element selection operator, also expressed using square brackets. Unlike a list literal, a square-brackets expression directly following another expression does not evaluate to a list value, but instead selects an element from the value of the preceding expression.

```
lst[0]
```

```
10
```

```
lst[1]
```

```
20
```

In both the example mentioned above mathematically we can represent list similar to a set.

$lst[(0, 10), (1, 20)]$ – where



5. Identify Which of the following are List, Tuple and class?

- | | | |
|---------------------------------------|---|-------|
| a) arr [1, 2, 34] | - | List |
| b) arr (1, 2, 34) | - | Tuple |
| c) student [rno, name, mark] | - | Class |
| d) day= ('sun', 'mon', 'tue', 'wed')- | | Tuple |

- e) x= [2, 5, 6.5, [5, 6], 8.2] - List
f) employee [eno, ename, esal, eaddress] – Class

PART IV

Answer the following questions (5Marks)

1. How will you facilitate data abstraction? Explain it with suitable example.

To facilitate data abstraction, we need to create two types of functions: constructors and selectors.

Constructors are functions that build the abstract data type.

Selectors are functions that retrieve information from the data type.

For example, we have an abstract data type called city. This city object will hold the city's name, and its latitude and longitude. To create a city object, we use a function like

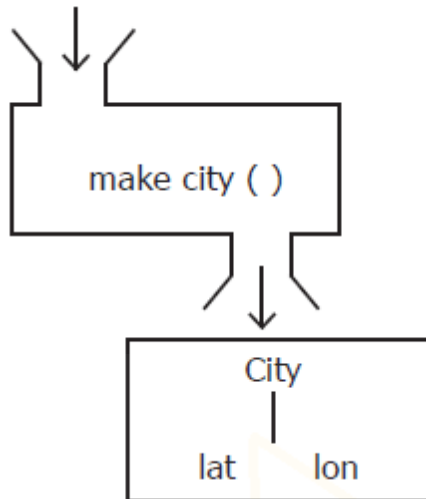
getname(city)

getlat(city)

getlon(city)

Here makecity (name, lat, lon) is the constructor which creates the object city.

(name, lat, lon) → value passed as parameter

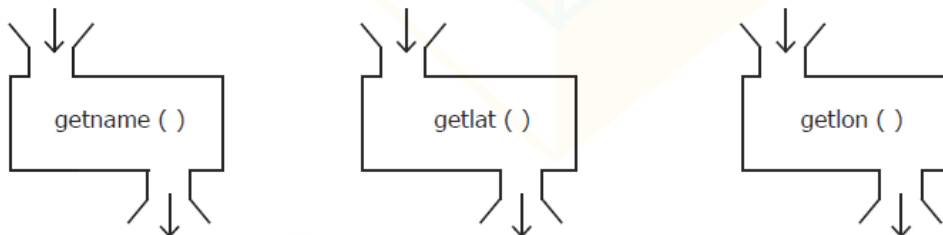


Selectors are nothing but the functions that retrieve information from the data type. Therefore in the above code

```

getname(city)
getlat(city)
getlon(city)
  
```

city → value passed as parameter city → value passed as parameter city → value passed as parameter



2. What is a List? Why List can be called as **Pairs**? Explain with suitable example.

List is constructed by placing expressions within square brackets separated by commas. Such an expression is called a list literal.

List can store multiple values.

Each value can be of any type and can even be another list.

Example for List is [10, 20].

Any way of bundling two values together into one can be considered as a pair. Lists are a common method to do so. Therefore List can be called as Pairs.

The elements of a list can be accessed in two ways.

The first way is via our familiar method of multiple assignments.

Example:

```
lst := [10, 20]
x, y := lst
```

In the above example x will become 10 and y will become 20.

A second method for accessing the elements in a list is by the element selection operator, also expressed using square brackets.

Example:

```
lst[0]
10
lst[1]
20
```

Representing Rational Numbers Using List :

We can represent a rational number as a pair of two integers in pseudo code: a numerator and a denominator.

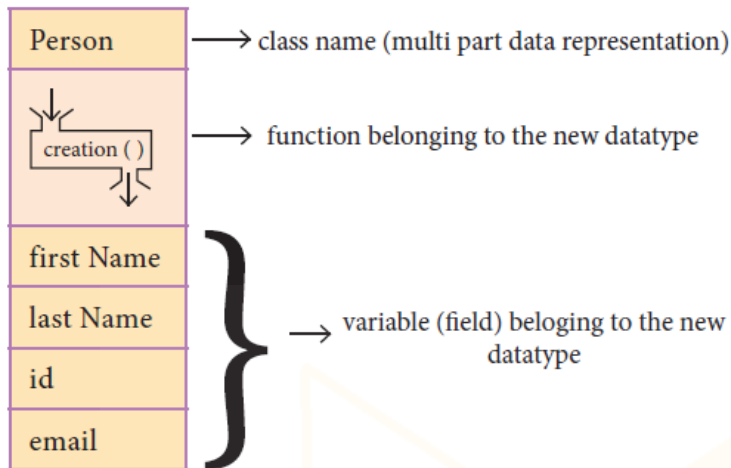
```
rational(n, d):
return [n, d]
numer(x):
return x[0]
denom(x):
return x[1]
```

3. How will you access the multi-item? Explain with example.

We can use the structure construct (In OOP languages it's called class construct) to represent multi-part objects where each part is named (given a name). Consider the following pseudo code:

```
class Person:
creation( )
firstName := " "
lastName := " "
id := " "
email := " "
```

The new data type Person is pictorially represented as



Let main() contains	
p1:=Person()	statement creates the object.
firstName := " Padmashri "	setting a field called firstName with value Padmashri
lastName := "Baskar"	setting a field called lastName with value Baskar
id := "994-222-1234"	setting a field called id value 994-222-1234
email="compsci@gmail.com"	setting a field called email with value compsci@gmail.com
- - output of firstName : Padmashri	

The class (structure) construct defines the form for multi-part objects that represent a person. Its definition adds a new data type, in this case a type named Person. Once defined, we can create new variables (instances) of the type.

In this example Person is referred to as a class or a type, while p1 is referred to as an object or an instance. We can think of class Person as a cookie cutter, and p1 as a particular cookie. Using the cookie cutter we can make many cookies. Same way using class we can create many objects of that type.

Prepared By

A.BASEERA NASHRIN , M.Sc., B.Ed.,

nashrinbaseera@gmail.com