પુ·ના International School

Shree Swaminarayan Gurukul, Zundal

Term – II Class – XI Computer Science

Dictionaries

Short Answer type Questions [2 mark each]

Question 1:
How are dictionaries different from lists ?
Answer:
The dictionary is similar to lists in the sense that it is also a collection of data-items just like lists but it is different form lists in the sense that lists are sequential collections and dictionaries are non-sequential collections. In lists, where there is an order associated with the data items because they act as storage units for other objects or variables created. Dictionaries are different from lists and tuples because the group of objects they hold are not in any particular order, but rather each object has its own unique name, commonly known as a key.

Question 2:
When are dictionaries more useful than lists ?
Answer:
Dictionaries can be much more useful than lists. For example : suppose we wanted to store all our friend's cell phone numbers. We could create a list of pairs (phone number, name), but once this list becomes long enough searching this list for a specific phone number will get time consuming. Better would be if we could index the list by our friend's name. This is precisely what a dictionary does.

Question 3:
Write a program to print the worker's informations (Name age, salary) in records format.
Answer:

```
Employees = {'Rohan' : {'age' : 20, 'salary' : 10000}, 'Mohan' : {'age' : 30, 'salary' : 15000}}
for key in Employees :
print "Employee", key + ':'
print 'Age :' + str (Employees [key] ['age']
print 'Salary :' + str (Employees [key] ['salary'])
```

Question 4:
What is a key-value pair with reference to Python dictionary ?
Answer:
A dictionary is a mapping between a set of indices (which are called keys) and a set of values. Each key maps a value. The association of a key and a value is called a key-value pair.

Question 5:
What are the characteristics of Python Dictionaries ?
Answer:
The 3 main characteristics of a dictionary are :

1. Dictionaries are Unordered : The dictionary elements (key-value pairs) are not in ordered form.
2. Dictionary Keys are Case Sensitive : The same key name but with different case are treated as different keys in Python dictionaries.
3. No duplicate key is allowed : When duplicate keys encountered during assignment, the last assignment wins.
4. Keys must be immutable : We can use strings, numbers or tuples as dictionary keys but something like [‘key’] is not allowed.

Long Answer type Questions [4 mark each]

Question 1:
Explain cmp(dict1, dict2) with example.
Answer:
Description
The method cmp( ) compares two dictionaries
based on key and values.
Syntax
Following is the syntax for cmp( ) method :
cmp(dict1, dict2)
Parameters
• dict1 – This is the first dictionary to be compared with dict2.
• dict2 – This is the second dictionary to be compared with dict1.
Return value :
This method returns 10 if both dictionaries are equal, – 1 if dictl < dict2 and 1 if dictl > dict2.
Example :
The following example shows the usage of comp( ) method.

```
# !/user/bin/python
dict1 = {‘Name’ : ‘Zara’, ‘Age’ : 7};
dict2 = {‘Name1 : ‘Mahnaz’, ‘Age’ : 27};
dict3 = {‘Name’ : ‘Abid’, ‘Age’ : 27);
dict4 = {‘Name’ : ‘Zara’, ‘Age’ : 7};
print “Return Value : %d”, % cmp (dict1, dict2)
print “Return Value : %d”, % cmp (dict2, dict3)
print “Return Value : %d”, % cmp (dictl, dict4)
```

Let us compile and run the above program, this will produce the following result :

```
Return Value : – 1    .
Return Value : 1
Return Value : 0
```

Question 2:
Explain str(dict) with example.
Answer:

Description
The method str( ) produces a printable string representation of a dictionary.
Syntax
Following is the syntax for str( ) method :
str(dict)
Parameters
• diet – This is the dictionary.
Return Value
This method returns string representation.
Example :
The following examples shows the usage of str() method.
# !/user/bin/python
diet = {'Name' :'Zara', 'Age' : 7);
print "Equivalent String : %s" %str (dict)
Let us compile and run the above program, this will produce the following result :
Equivalent String : {'Age' : 7, 'Name' : 'Zara'}

Question 3:
Describe dict.fromkeys( ) with example.
Answer:
dict.fromkeys( )
The method fromkeys( ) creates a new dictionary with keys from seq and values set to value.
Syntax
Following is the syntax for fromkeys( ) method :
dict.fromkeys{seq[,value])}
Parameters :
- seq – This is the list of values which would be used be used for dictionary keys preparation.
- value – This is optional, if provided then value would be set to this value.

Return Value
This method returns the list.
Example :
The following example shows the usage of fromkeys( ) method.
#!/user/bin/python
seq = {'name', 'age', 'sex')
diet = dict.from keys(seq)
print "New Dictionary : %s" % str(dict)
dict = dict.from keys(seq, 10)
print "New Dictionary : %s" % str(dict)
Let us compile and run the above program, this will produce the following result :
New Dictionary : {'age' : None, 'name' : None, 'sex': None}
New Dictionary : {'age' : 10, 'name' : 10, 'sex' : 10}

Question 4:
Write a program to input total number of sections and class teacher's name in 11th class and display all information on the output screen.
Answer:
class xi = diet ( )

```
n = input (“Enter total number of section in xi class”)
i = 1
while i < = n :
a = raw_input (“enter section”)
b = raw input (“enter class teacher name”)
classxi[a] = b
i = i + 1
print “Class”, “\t”, “Section”,
\t„, “Teacher Name”
for i in classxi :
print “XI”, “\t”, i, “\t”, class xi[i]
```

Question 5:

Write a Python program to input ‘n’ names and phone numbers to store it in a dictionary and to input any name and to print the phone number of the particular name.

Answer:

```
phonebook = dict( )
n = input (“Enter total number of friends”)
i = 1
while i<=n :
a = raw_input (“Enter name”)
b = raw_input (“Enter phone number”)
phonebook [a] = b
i = i + 1
name = raw_input (“Enter name”)
f = 0
l = phonebook.keys( )
for i in l :
if (comp (i, name) = = 0) :
print “Phone number =”, phonebook[i]
f = 1
if (f = = 0):
print “Given name not exist”
```

Question 6:

Write a Python program to input ‘n’names and phone numbers to store it in a dictionary and to search and print the phone number of that particular name.

Answer:

```
phonebook=dict( )
n=input(” Enter total number of friends to create a telephone book”)
i=l
while i<=n:
a=raw_input(“enter name”)
b=raw_input(“enter phone number”)
phonebook[a]=b
i=i+l
name=raw_input(“enter friend’s name to search for”)
f=0
```

```
l=phonebook.keys( )
for i in l:
if(cmp(i,name) == 0):
print “Phone number= “,phonebook[i]
f=l
if (f==0):
print “Given name not exist”
```

Output:

```
Enter total number of friends to create a telephone
book 5
enter name Ramu
enter phone number 9842311111
enter name Raju
enter phone number 9842309564
enter name Sarita
enter phone number 9842398765
enter name Sowmya
enter phone number 9842399922
enter name Karan
enter phone number 9842366554
enter friend’s name to search for Sarita
Phone number = 9842398765
```

Question 7:

Write the code to input any 5 years and the population of any city and print it on the screen.

Answer:

```
city=dict( )
n=5
i=l
while i<=n:
a=raw_input(” enter city name”)
b=raw_input(“enter population”)
city[a] =b
i=i+l
print city
```

Output :

```
enter city name Chennai
enter population 3400000
enter city name Mumbai
enter population 8900000
enter city name Bangalore
enter population 98700
enter city name Calicut
enter population 560000
enter city name Hyderabad
enter population 900000
{‘Bangalore’: ‘ 98700’/ Hyderabad’: ‘ 900000’, ‘ Chennai’: ‘ 3400000’, ‘ Mumbai’: ‘
8900000’, ‘ Calicut’: ‘ 560000’}
```

Question 8:
Write a code to create customer's list with their number & name and delete any particular customer using his /her number.

Answer:

```
customer=dict( )
n=input(" Enter total number of customers")
i=l
while i<=n:
a=raw_input("enter customer name")
b=raw_input("enter customer number")
customer[a]=b
i=i+l
name=raw_input(" enter customer name to delete")
del customer[name]
l=customer.keys( )
print "Customer Information"
print "Name"/\t'/'number"
for i in 1:
print i/\t',customer[i]
```

Output :

```
Enter total number of customers 5
enter customer name Rohan
enter customer number 123678
enter customer name Ranjish
enter customer number 123677
enter customer name Suraj
enter customer number 123899
enter customer name Damu
enter customer number 123777
enter customer name Dhariya
enter customer number .123885
enter customer name to delete Suraj
Customer Information
Name number
Rohan    123678
Dhariya  123885
Ranjish    123677
Damu    123777
```

Question 9:
Find errors from the following codes:

```
c=dict( )
n=input(Enter total number) i=l
while i<=n
a=raw_input("enter place")
b=raw_input(" enter number")
c(a)=b
i=i+l
print "place","\t"/'number"
for i in c:
```

print i,”\t”,cla[i]

Answer:

```
c=dict()
n=input(“Enter total number” )
i=l
while i<=n:
a=raw_input(“enter place”)
b=raw_input(“enter number”)
c[a]=b
i=i+l
print “place”,”\t”/”number”
for i in c:
print i.”\t”.c[il
```