**MCQ**

1. **What will be the output of the following code snippet?**

def func(a,b):

return b if a ==0 else func(b%a, a)

print(func(30,75))

1. 10
2. 20
3. 15
4. 0

Ans: (c) 15

1. **numbers=(4,7,19,2,89,45,72,22)**

**sorted\_numbers=sorted(numbers)**

**even=lambda a:a%2==0**

**even\_numbers=filter(even, sorted\_numbers)**

**print(type(even\_numbers))**

1. **Int**
2. **Filter**
3. **List**
4. **Tuple**

**Ans: (b) Filter.**

1. **As what datatype are the \*args stored, when passed into**
2. **Tuple**
3. **List**
4. **Dictionary**
5. **None**

**Ans: (a) Tuple**

1. **set1={14,3,55}**

**set2={82,49,62}**

**set3={99,22,17}**

**print(len(set1+set2+set3))**

1. **105**
2. **270**
3. **0**
4. **Error**

**Ans: (d) Error**

1. **Which keyword is used in Python to raise expectations?**
2. **raise**
3. **try**
4. **goto**
5. **expect**

**Ans: (a) raise**

1. **Which of the following modules need to be imported to handle date time computations in Python?**
2. **timedate**
3. **date**
4. **datetime**
5. **time**

**Ans: (c) datetime**

1. **What will be the output of the following code snippet?**

**print(4\*\*3+(7+5)\*\*(1+1))**

1. **248**
2. **169**
3. **208**
4. **233**

**Ans: (c) 208**

1. **Which of the following functions converts date to corresponding time in python?**
2. **strptime**
3. **strftime**
4. **both a) and b)**
5. **none**

**Ans: (a) strptime**

1. **The python tuple is \_\_\_\_\_\_ in nature.**
2. **mutable**
3. **immutable**
4. **unchangeable**
5. **done.**

**Ans: (b) immutable**

1. **The \_\_\_\_ is a built – in function that returns a range object that consists series of integer numbers, which we can iterate using for loop.**
2. **range()**
3. **set()**
4. **dictionary{}**
5. **none of the mention above.**

**Ans: (a) range()**

1. **Amongst which of the following is a function which does not have any name?**
2. **Del function**
3. **Show function**
4. **Lambda function**
5. **None of the mentioned above**

**Ans: C. Lambda function**

1. **The module pickle is used to\_\_\_\_\_\_\_**
2. **Serializing Python object structure**
3. **De-serializing Python object structure**
4. **Both A and B**
5. **None of the mentioned above**

**Ans: C. Both A and B**

1. **Amongst which of the following is/are the method of convert python objects for writing data in a binary file?**
2. **set()method**
3. **dump()method**
4. **load()method**
5. **None of the mentioned above**

**Ans: B. dump()method**

1. **Amongst which of the following is/are the method used to unpickling data from a binary file?**
2. **load()method**
3. **set()method**
4. **dump()method**
5. **None of the mentioned above**

**Ans: A. load()method**

1. **A text file contains only textual information consisting of\_\_\_\_\_**
2. **Alphabets**
3. **Numbers**
4. **Special symbols**
5. **All of the mentioned above**

**Ans: D. All of the mentioned above**

1. **Which Python code could replace the ellipsis(…) below to get the following output? (Select all that apply.)**

**captains={“Enterprise” : “Picard”,**

**“Voyager” : “Janeway”,**

**“Defiant” : “Sisko”,**

**}**

**Enterprise Picard**

**Voyager Janeway**

**Defiant Sisko**

1. **for ship, captain in captains.item():**

**print( ship, captain)**

1. **for ship, in captains:**

**print(ship, captains[ship])**

1. **for ship in captains:**

**print(ship, captains)**

1. **both a and b**

**Ans: a) for ship, captain in captains.item():**

**print(ship, captain)**

1. **Which of the following lines of code will create an empty dictionary named captains?**
2. **captains = {dict}**
3. **type(captains)**
4. **captains.dict()**
5. **captains = {}**

**Ans: d) captains = {}**

1. **Now you have your empty dictionary named captains. It’s time to add some data!**

**Specially, you want to add the key-value pairs “Enterprise : “Picard”,**

**“Voyager” : “Janeway” and “Defiant” : “Sisko”.**

**Which of the following code snippets will successfully add these key-value pairs to the existing captains dictionary?**

1. **captains {“Enterprise” = “Picard”}**

**captains {“Voyager” = “Janeway”}**

**captains {“Defiant” = “Sisko”}**

1. **captains[“Enterprise”] = “Picard”**

**captains[“Voyager”] = “Janeway”**

**captains[“Defiant”] = “Sisko”**

1. **captains = {“Enterprise” : “Picard”, “Voyager” : “Janeway”,**

**“Defiant” : “Sisko”}**

1. **None of the above**

**Ans: c) captains = {“Enterprise” : “Picard”, “Voyager” : “Janeway”,**

**“Defiant” : “Sisko”}**

1. **You’re really building out the Federation Starfleet now! Here’s what you have:**

**captains = {“Enterprise” : “Picard”, “Voyager” : “Janeway”, “Defiant” : Sisko”, “Discovery” : “unknown”}**

**Now, say you want to display the ship and captain names contained in the dictionary, but you also want to provide some additional context. How could you do it?**

1. **for item in captains.items():**

**print(f “The[ship] is captained by [captain].”)**

1. **for ship, captain in captains.items():**

**print(f “The{ship} is captained by {captain}.”)**

1. **for captain, ship in captains.items():**

**print(f “The{ship} is captained by {captain}.”)**

1. **All are correct**

**Ans: d) All are correct**

1. **You’ve created a dictionary, added data, checked for the existence of keys, and iterated over it with a for loop. Now you’re ready to delete a key from this dictionary:**

**captains = {“Enterprise” : “Picard”, “Voyager” : “Janeway”,**

**“Defiant” : “Sisko”, “Discovery” : “unknown”}**

**What statement will remove the entry for the key “Discovery”?**

1. **del captains**
2. **captains.remove()**
3. **del captains[“Discovery”]**
4. **captains[“Discovery”].pop()**

**Ans: c) del captains[“Discover”]**