

Unique Paper Code : 42341102
Name of the Course : B.Sc. Prog./Mathematical Sc
Name of the paper : Problem Solving with Computers
Semester : I

Duration: 3 Hours

Maximum Marks: 75

Instructions for Candidates:

Attempt any Four Questions. All Questions carry equal marks.

- Q1
- Give the output that will be produced on execution of the following code segments:

```
o x=3
  y=2
  if x>2:
    if y>2:
      z = x+y
      print("Z is: " , z)
    else:
      print("x is: ", x)
```

```
o for i in range(1,5):
    j = 0
    while j<i:
      print(j, end=" ")
      j += 1
```

```
o d={'Name':'Alice', 'Age':7}
  print(d.get('Name'))
  print(d.get(1, 'Invalid'))
```

```
o float(4+int(2.39)%2)
```

```
o "asdf"[::-1]
```

```
o print(list('Hello'))
```

- Write a function `series_sum(n)` in Python to calculate the sum of the first n terms of the following series:

$1/2 - 1/4 + 1/8 - 1/16 + 1/32 - 1/64 + \dots$

Q2

- Write Python statements to read name and age as input from a user and print the year in which the user turns 100 years old.
- Give the output that will be produced on execution of the following code segments:

- ```
l1= [10, 22, 'x', 'y', 33, 44]
print(l1[3:])
```

- ```
max(l1)
```

- ```
type(l1[2])
```

- ```
l2=[4,5]
l1.extend(l2)
```

- ```
l1.reverse()
```

- ```
try:
    f = open("MyFile.txt", "r")
    f.write("This is my file")
except IOError:
    print("Cant open file")
else:
    print("Content writen")
```

- ```
list=['a',0,2]
for x in list:
 try:
 print("The value is ", x)
 r=1/int(x)
 break
 except Exception as e:
 print(e.class, " occured")
 print("Next value")
 print()
print("Reciprocal of ", x, " is ", r)
```

- ```
Word1= " Hello first year students"
Word2=" Hello second year students"
for i in Word1:
    if i in Word2:
        print(i , end=" ")
```

Q3

- Write statements to create a file `Countries.txt` with the following rows:

```
$India$USA$Nepal$
$Indonasia$Ireland$
$Srilanka$Russia$
```

Consider the file `Countries.txt`. Give the output that will be produced on execution of the following code segment:

```
f1 = open("Countries.txt", "r")
name=f1.readline().strip("$\n")
while name:
    if name.startswith("I"):
        print(name)
    else:
        pass
    name=f1.readline().strip("$\n")
```

- Give the output that will be produced on execution of the following code :

```
def f():
    try:
        s="abc"
        print(s[3])
    except ZeroDivisionError:
        print("Divided by zero")

def main():
    try:
        f()
        print("After the function call")
    except IndexError:
        print("Index out of bound")
    except:
        print("Exception in main")

main()
```

Q4

Apply Insertion sort scheme of sorting on the following list to sort it in ascending order:

```
lst=[5, 4, 3, 11, 14, 2, 6, 7]
```

Show the list after each iteration.

How many iterations are required to sort the above list?

Apply Binary Search to search for the item 9 in the sorted list. At each step, show the index at which the value is compared with 9.

Under what circumstances, you would prefer to use *linear search* over *binary search*? Justify your answer.

Q5

- Define a class `Rectangle` having `length` and `breadth` of rectangle as the data members and the methods to do the following:

Methods:

- Constructor to initialize the data members `length` and `breadth`.
- `area()` to calculate area of the rectangle.
- `perimeter()` to calculate perimeter of the rectangle.
- `__str__` to return string representation for displaying the data members suitably.

Also, write Python statements to:

- Create an object of class `Rectangle` of length 4 and breadth 3.
 - Invoke the method `area()`.
 - Invoke the method `perimeter()`.
 - Print all the data members of the class.
- Give the output that will be produced on execution of the following code segment:

```
class Abc:
    const=9
    def __init__(self,name,id):
        self.name = name
        self.id = id

Abc.const = 99
A = Abc("John",123)
B = Abc("Diana",444)
b.const = 9
c = Abc("William",222)

print("a :", a.const, "b :", b.const, "c :",c.const)
```

Q6

- Consider the following two sets:

```
setx = set(["green", "blue", "yellow", "red"])
sety = set(["blue", "yellow", "pink", "orange"])
```

Write the Python statements for each of the following operations:

- Adding an element to the set `setx`.
- Compute `xUy` as union of `setx` and `sety`.
- Compute `xminusy` as set difference between `setx` and `sety`
- Remove "blue" from `setx`

- Consider the following string:

```
greeting = "Good Morning. Have a Good Day!! "
```

Give the output for the following function calls:

- `greeting.find("a")`
- `greeting.swapcase()`
- `greeting.istitle()`
- `greeting.replace("Good", "Sweet")`
- `greeting.strip()`
- `greeting.endswith("!!")`

- Consider the tuple `t1` defined below :

```
t1 = (12, 5, 2, 4, 17, 44, 7, 6, 9, 10).
```

Write a Python statement to:

- Print first half of the values of `t1` in one line and the other half in another line.
- Produce a list comprising all the even numbers in the tuple `t1`.