VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : PP LAB

Name K. Sree Indira Sivani Roll No. 1602-21-733-052 Page No. 119

* PRELAB QUESTIONS-10

1) What is pickling in python?

A: The process to convert any python object (list, tuple, & dictionary) into byte streams is called pickling (or) flattening (or) marshalling.

2) Give examples on the usage of try, except in python for exception handling.

A:
```
try:
    x=int(input("Enter a no:"))
    y=int(input("Enter a no:"))
    q=x/y
    print(q)
except ZeroDivisionError:
        print("Mathematical error")
else:
        print("No exception")
```

3) Why exception handling is required in programming?

A: An exception is an event which occurs during execution of program and disturbs the normal flow of program instructions. So it is necessary to handle the exceptions using try & except blocks.

4) How to raise an exception in python? Give an example:

A:
```
try:
    n=int(input())
    p=int(input())
    print(n**p)
    raise TypeError("hello").
except TypeError ast:
        print("Enter a valid integer:")
```

* Raising an exception (or) instantiating an exception is generally used for raising exception in case of user defined exceptions.

5) How does try-except-else clause work in python? Give example:

A:
* The statements mentioned in try block raises an exception then the except block will be executed and else block will not.
* If the try block do not raise an exception; then after executing try; else block is executed.

```
try:
    x= int(input("Enter no:"))
except TypeError:
    print("Enter only an integer:")
else:
    print("No exception")
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : PP LAB

Name K. Sree Indira Sivani Roll No. 1602-21-733-052 Page No. 121

PRELAB PROGRAMS: 10

1) Python program to create a file containing 10 numbers. Read the contents of file and display the square of each number.

```
import pickle
l=[]
for i in range(10):
    x=int(input())
    l.append(x)
f=open("numbers.txt","wb")
pickle.dump(l,f)
f.close()
f=open("numbers.txt","rb")
y=pickle.load(f)
f.close(); s=[]
for i in y:
    m=i*i
    s.append(m)
print("Squares=",s)
```

Output:

```
1
10
12
7
8
9
6
16
15
20
Squares=
[1,100,144,49,
64,81,36,
256,225,400]
```

2) Write a program to handle exceptions from an invoked function:

```
def divide(num,den):
    try:
```

```
        q=num/den
    except ZeroDivisionError:
        print("Zero cannot be denominator")
divide(10,0)
```

O/P:

Zero cannot be denominator.

3) Write a program to store the string in a file using file pickling and then read the contents of the file and display it on the monitor:

```
import pickle
x=input("Enter a string:")
f=open("string.txt","wb")
pickle.dump(x,f)
f.close()
f=open("string.txt","rb")
x=pickle.load(f)
f.close()
print("String:",x)
```

Output:

Enter a string: Python Programming

String: Python Programming.

4) Write a program that opens a file and writes data to it. Handle exceptions that can be generated during the I/O operations.

```
try:
    f = open("append.txt", "a")
    f.write("Exception Handling"); f.read()
    f.close()
except IOError:
        print ("File does not exist").
```

Output:

"File is not present"

5) Write a program to accept 2 numbers and divide one number by another. Handle ZeroDivisionError exception.

```
try:
    num = int(input("Enter numerator:"))
    den = int(input("Enter denominator:"))
    quotient = num/den ; print(q)
except ZeroDivisionError:
    print("Denominator can't be zero")
```

Output:

Enter numerator: 10
Enter denominator: 0
Denominator can't be zero.

PRELAB PROGRAMS: 10

6) Write a program having finally block to raise the exception that will be defined by an outer try except block.

```
try:
    f=open("file1.txt","r")
    try:
        print(f.read())
    except IOError:
        print("Error")
    else:
        print("No exception")
    finally:
        f.close()
except FileNotFoundError:
    print("No such file")
```

Output:

```
hello
No exception
```

Output:

```
Enter the no. of students:2
Roll no: 52
Name: Sivani
Gender: f
Age: 18
Roll no:41
Name: Sarayu
Gender: f
Age: 19
{52:['sivani', 'f', 18], 41:['sarayu', 'f', 19]}
Enter a file name: detail.txt
Enter the roll no to print details: 41
['sarayu', 'f', 19].
```

VASAVI COLLEGE OF ENGINEERING
(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : PP LAB

Name K. Sree Indira Sivani Roll No. 1602-21-733-052 Page No. 125

LAB PROGRAMS - 10

1) Program to create a dictionary of student details and store it in the file. Then load the details and details of Rno:10.

```
import pickle
d={}
n=int(input("Enter the no. of students:"))
for i in range(n):
        n=int(input("Roll no:"))
        l=[]
        m=input("Name:"); l.append(m)
        x=input("Gender:"); l.append(x)
        a=int(input("Age:")); l.append(a)
        d[n]=l
print(d)
f=open("detail.txt","wb")
pickle.dump(d,f)
f.close()
f=open("detail.txt","rb")
p=pickle.load(f)
f.close()
y=int(input("Enter the roll no. to print details:"))
print(p[y])
```

VASAVI COLLEGE OF ENGINEERING

(AUTONOMOUS)
(Affiliated to Osmania University)
Hyderabad - 500 031.

DEPARTMENT OF : CSE

NAME OF THE LABORATORY : PP LAB

Name K. Sree Indira Sivani Roll No. 1602-21-733-052 Page No. 126

2) Program to create a list of cubes of the first 10 natural numbers and store it in a file. Read the data from the files and display the even indexed elements.

Output:

```
* import pickle
l=[i**3 for i in range(1,11)]
f=open("cubes.txt","wb")
pickle.dump(l,f)
f.close()
f=open("cubes.txt","rb")
x=pickle.load(f)
f.close()
e=[]
for i in range(len(x)):
    if i%2==0:
        e.append(x[i])
print("Even indexed:",e)
```

Even indexed: [1, 27, 125, 343, 729]

3) Write a program that prompts the user to enter a number. If the number is positive or zero print it; otherwise raise an ValueError exception:

```
try:
    n=int(input("Enter an integer:"))
    if n<0:
        raise ValueError
    elif n>0:
        print("Positive Number")
    else:
        print("Zero")
```

Output:

1) Enter an integer: 5
Positive number
2) Enter an integer: -7
Negative number not allowed
3) Enter an integer: 0
Zero

```
except ValueError:
    print("Negative number not allowed:")
```

4) Write a program that ~~validates name~~ infinitely prints natural numbers. Raise the StopIteration after displaying first 20 natural numbers to exit from the program.

```
try:
    n=int(input("Enter the no. of numbers: "))
    i=1;c=0
    while(n!=0):
        print(i)
        c+=1
        i+=1
        if c==20
            raise StopIteration

except StopIteration:
    print("Exceeded limit.")
```

Output:

```
Enter no. of nos:5
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

5) Write a program that validates name and age as entered by the user to determine whether the person can cast the vote or not. Define 2 user defined classes – invalidAge, invalidName.

```
class invalidAge(Exception):
    def display(self):
        print("Age should be >18")
class invalidName(Exception):
    def display(self):
        print("Invalid Name:")
try:
    n=input("Enter your name:")
    if n.isalpha():
        print("Valid name")
    else:
        raise invalidName
    x=int(input("Enter your age:"))
    if x>18:
        print("Eligible to vote")
    else:
        raise invalidAge
except invalidAge as ia:
        ia.display()
except invalidName as iv:
        iv.display().
```

Output:

```
Enter your name:
sivani7
Invalid name.
Enter your name:
sivani
Valid name
Enter your age:17
Age should be >18.
```

18/8/22