

Department of Computer Science
Mid Semester Assignment Summer 2020

Subject:

Data

Sciences

Time: 90 mins

BS (CS,SE)

Name: Muhammad Hamza

ID - 13136

Instructor: M.Ayub Khan

There are total **3** questions in this paper.

Max Marks: 30

Note:

At the top of the answer sheet there must be the ID, Name and semester of the concerned Student.

Students must have to provide the output of their respective programs. Students have same answers or programs will be considered fail. Programs in Python and codes should be explained clearly.

**Each question carry equal marks.
Please answer briefly.**

Q1. What type of errors do occur in Python, write the a program with different types of errors as well as write separate correction code?

Ans.

Explanation:

1) Syntax Error: Syntax errors are the most basic type of error. They arise when the Python parser is unable to understand a line of code. Syntax errors are almost always fatal, i.e. there is almost never a way to successfully execute a piece of code containing syntax errors. Some syntax errors can be caught and handled, like `eval("")`, but these are rare.

When the rules of the coding are violated. Like, here in this code, Indentation is not provided after if statement, which led to an error in the code.

```
#Syntax Error
num=int(input('Enter Number: '))
if(num%2==0):
print("Even Number")
else:
print("Odd Number")
```

```
File "<ipython-input-3-285f8755661b>", line 4
    print("Even Number")
    ^
```

IndentationError: expected an indented block

```
#Correct Code
num=int(input('Enter Number: '))
if(num%2==0):
    print("Even Number")
else:
    print("Odd Number")
```

```
Enter Number: 12
Even Number
```

2) Logical Errors: These are the most difficult type of error to find, because they will give unpredictable results and may crash your program. A lot of different things can happen if you have a logical error. However, these are very easy to fix as you can use a debugger, which will run through the program and fix any problems.

These errors generally occur due to some wrong logical value, processed during the execution. Like, in here, we have divided 12, with 0, which caused an error here.

```
#Logical Error
n1=int(input('Enter number 1: '))
n2=int(input('Enter number 2: '))
print(n1/n2)
```

```
Enter number 1: 12
Enter number 2: 0
```

```
-----
ZeroDivisionError                                Traceback (most recent call last)
<ipython-input-5-06773d31a0c7> in <module>
      2 n1=int(input('Enter number 1: '))
      3 n2=int(input('Enter number 2: '))
----> 4 print(n1/n2)
```

ZeroDivisionError: division by zero

Such errors can easily be handled with Exception handling.

```
#Correct Code
try:
    n1=int(input('Enter number 1: '))
    n2=int(input('Enter number 2: '))
    if(n1==0 or n2==0):
        raise NumZero
except NumZero:
    print("Number Cannot be Zero!")
print(n1/n2)
```

```
Enter number 1: 12
Enter number 2: 0
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-11-e400013529f1> in <module>
      5     if(n1==0 or n2==0):
----> 6         raise NumZero
      7 except NumZero:
```

NameError: name 'NumZero' is not defined

During handling of the above exception, another exception occurred:

```
NameError                                Traceback (most recent call last)
<ipython-input-11-e400013529f1> in <module>
      5     if(n1==0 or n2==0):
      6         raise NumZero
----> 7 except NumZero:
      8     print("Number Cannot be Zero!")
      9 print(n1/n2)
```

NameError: name 'NumZero' is not defined

Q2. What are Boolean String test, write the code for each Boolean string test code?

Ans.

Boolean string tests are the list of validation functions which will result as True or False for the check. The return type will be in Boolean value (True or False). Below are few of the examples of such test functions in Python are

```
my_string.isalnum() #check if all char are numbers
my_string.isalpha() #check if all char in the string are alphabetic
my_string.isdigit() #test if string contains digits
my_string.istitle() #test if string contains title words
my_string.isupper() #test if string contains upper case
my_string.islower() #test if string contains lower case
my_string.isspace() #test if string contains spaces
my_string.endswith('d') #test if string endswith a d
my_string.startswith('H') #test if string startswith H
```

To see what the return value (True or False) will be, simply print it out.

```
my_string="Hello World"
print my_string.isalnum() #False
print my_string.isalpha() #False
print my_string.isdigit() #False
print my_string.istitle() #True
print my_string.isupper() #False
print my_string.islower() #False
print my_string.isspace() #False
print my_string.endswith('d') #True
print my_string.startswith('H') #True
```

Q3. What is formatting string input mean in Python, write a program in which formatting string input is used?

Ans.

Explanation:

The string formatting input means to create new, formatted strings. The "%" operator is used to format a set of variables enclosed in a "tuple" (a fixed size list), together with a format string, The format() method allows you to format input selected parts of a string. Sometimes there are parts of a text that you do not control input, maybe they come from a database, or user input?

To control such values, add placeholders (curly brackets {}) in the text, and run the values through the format() input method

There are different options in Python to format a string. Some of them are given below:

```
#Foprmatting Input String
```

```
InputStr = "Hello World"
```

```
# Simple print the Input String  
print(InputStr)
```

```
# Print the Input String with 20 column size  
print("%20s"%InputStr)
```

```
# Print the Input String with left justified atribute with zero padding  
InpuStr = "this is string example....wow!!!";  
print (InpuStr.ljust(50, '0'))
```

```
# Print the Input String with right justified atribute with zero padding  
print (InpuStr.rjust(50, '0'))
```

Python Output:

```
>>>  
RESTART:  
E:\CDAC_31_12_2017_Backup\CheggIndia_Anshu_Office\PythonCode\CheggPython.py  
Hello World  
    Hello World  
>>>  
RESTART:  
E:\CDAC_31_12_2017_Backup\CheggIndia_Anshu_Office\PythonCode\CheggPython.py  
Hello World  
    Hello World  
this is string example....wow!!!00000000000000000000  
00000000000000000000this is string example....wow!!!  
>>>
```

