Topics: 1. Difference between C and C++ with Examples.

Answer)

Before differentiating between c and c++ let’s see the introduction of them :

C

C is a middle-level computer programming language which is developed by Dennis Ritchie in bells laboratory. only c language is known as a middle-level language among the generation of computer language because of its translation process. its translation process consists of two stages _

First stage :

It requires an interpreter, it translates files along with library functions only other lines remain ns same and an object file is created.

Second stage :

The object file is transferred to the compiler using a linker. The compiler compiles the remaining program and generates a .exe file.

C programming language is reliable, simple, and easy to use.

C++ :

C++ is an object-oriented programming language which is developed by **Bjarne Stroustrup** in bells laboratory. It is an advanced version of c . it is faster and simpler than C as it contains STL (standard template library). It is regarded high-level language. **C++ is intrinsically stingy with memory.**

SL NO	**C**	**C++**
1	It is a middle-level of language .	It is high-level language.
2	Here variable must be declared at the beginning along with datatype.	Variable can be declared anywhere.
3	It is a structural programming language.	It is an object-oriented programming language.
4	It is function-driven.	It is object-driven.
5	It does not support inheritance.	It supports single, multilevel inheritance .
6	It does not support operator overload.	It supports operator and function overload.
7	We can allocate memory dynamically using calloc() and malloc(). Suppose I want to allocate memory dynamically using malloc() then the syntax would be	It can be done using new() Syntax :

	javaTpoint=(cast-type*)malloc(size required)
	here javaTpoint = holds the address of the first byte of allocated memory.
	for example :
	javaTpoint=(int*)malloc(40*sizeof(int));
	
	javaTpoint holds the address of the first byte of allocated memory.
		Variable_type *name_of_variable=new Variable_type[length]
		Ex :
		Int *javaTpoint = new int[5];
		
		javaTpoint holds the address of the first byte of allocated memory.

8	It does not contain STL.	It contains STL such as vector, stack, ,queue and maps, etc.
9	Comparatively slow.	Faster than c.
10	Its extension is **.c** .	Its extension is **.cpp** .

Basic syntax difference between c and c++.

```
#include <stdio.h>
int main()
{
     int n; //declaring variable
    printf("enter your number : ");  //printing message
    scanf("%d",&n);      // for taking input
     printf("%d",n);      //printing output

    return 0;
}
```
// here we have to mention %d if the variable is int and %c if char and so on .. but in the case of c++ we have to declare it only once while variable declaration and then we do not have to be concerned about writing things as %d , %c or else for reading or printing.
	```
	#include <iostream>
	using namespace std;
	//in c++ we Can use #include<bits/stdc++.h> It includes all  STL.—but it consumes more time.

	int main()
	{
	   cout<<"enter the number : "<<endl; // //printing message
	   int n;       //declaring variable
	   cin>>n;    //for taking input
	   cout<<n;  //printing output


	    return 0;
	}
	```

2. Exception handling in Python.

Writing a program is easy. if syntax and logic are correct we think that that's it my program will work efficiently but there are some unexpected events such as—

Suppose you have written a program for finding the area of the square and if the length of the side given by the user is negative! Well, this is an exception because the length of the side cannot be negative. we can resolve this simply by adding a condition such as ---- if(length>0)

But this is a very simple example in complex programs we cannot simply use the if statement.

Why are we concerned with user input? Isn't it a user's problem?

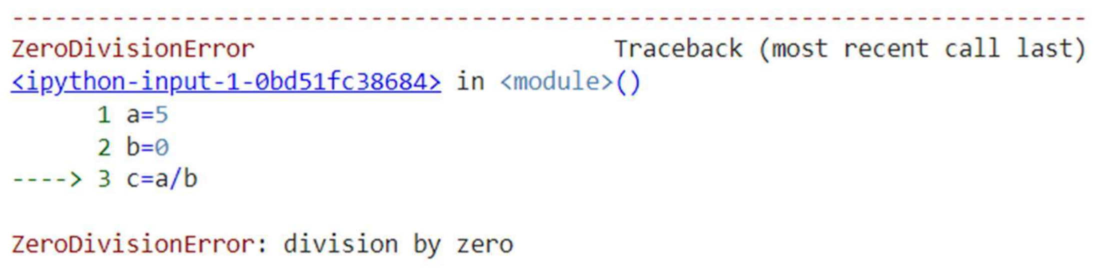
Well as a software engineer you must be concerned about what mistakes a user can commit. whenever there are runtime errors execution stops! this is a very big issue your whole software will stop working just because of a single runtime error.

Exception:

Exception generally occurs during the execution of the program and due to exception normal flow of instruction execution disrupts.

Different Exceptions in python :

ZeroDivisionError: division by zero :

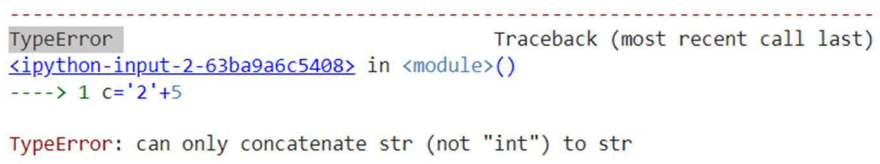


```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-1-0bd51fc38684> in <module>()
      1 a=5
      2 b=0
----> 3 c=a/b

ZeroDivisionError: division by zero
```

We know when we divide any number by zero the output is infinite so that is why this error appears.

TypeError :



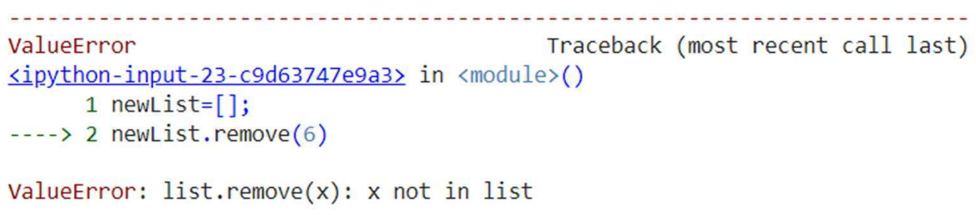
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-2-63ba9a6c5408> in <module>()
----> 1 c='2'+5

TypeError: can only concatenate str (not "int") to str
```

here '2' is a string and 2 is an integer and we are trying to evaluate two data types together thatswhy TypeError appears.

either both should be integer or string in order to get the result.

ValueError :



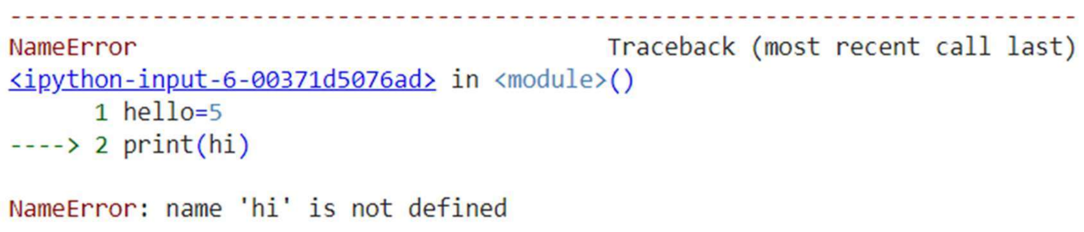
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-23-c9d63747e9a3> in <module>()
      1 newList=[];
----> 2 newList.remove(6)

ValueError: list.remove(x): x not in list
```

Here we are trying to remove an element that is not present in the list thatswhy value error occurred.

NameError :

output :

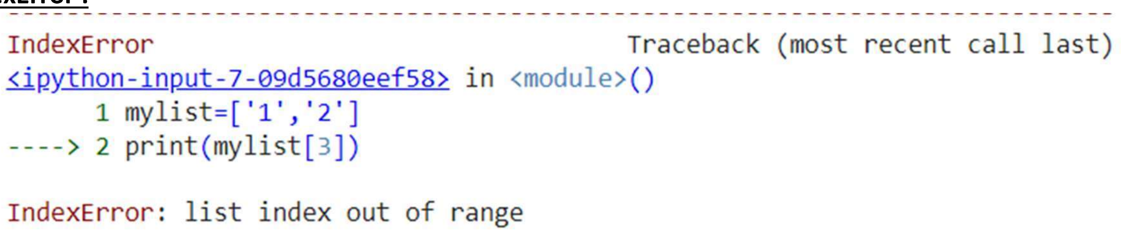


```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-6-00371d5076ad> in <module>()
      1 hello=5
----> 2 print(hi)

NameError: name 'hi' is not defined
```

Here we are trying to print a variable’s value that is not defined thatswhy NameError occurred.

IndexError :

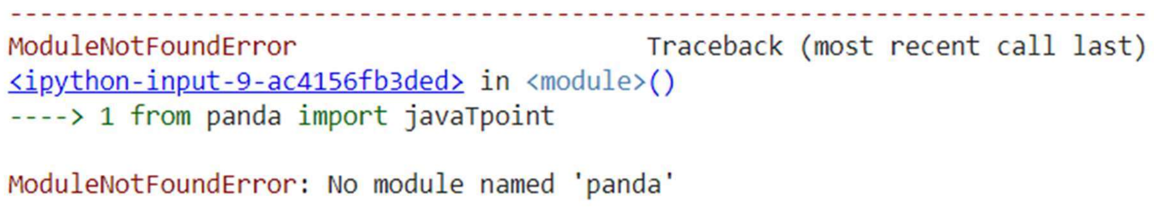


```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-7-09d5680eef58> in <module>()
      1 mylist=['1','2']
----> 2 print(mylist[3])

IndexError: list index out of range
```

Here we are trying to print an element of mylist through its index which does not exist thatswhy IndexError occurred.

ModuleNotFoundError :

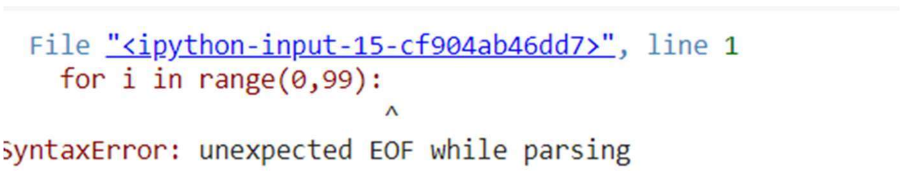


```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-9-ac4156fb3ded> in <module>()
----> 1 from panda import javaTpoint

ModuleNotFoundError: No module named 'panda'
```

There is no such module as panda.

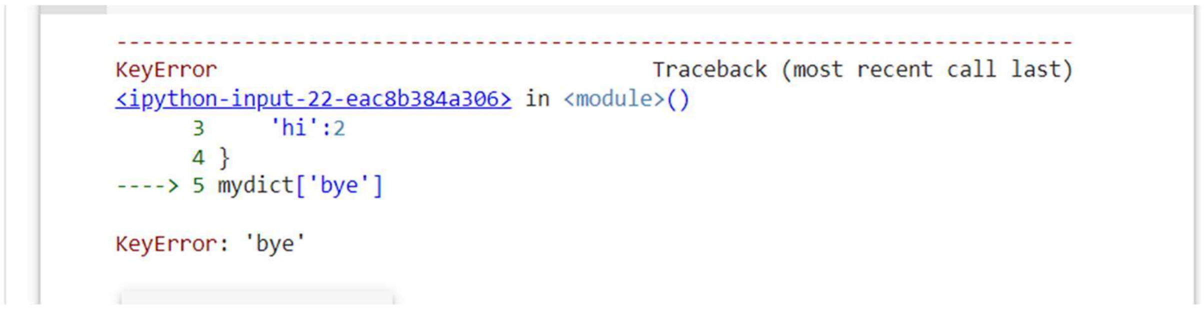
EOF :



```
  File "<ipython-input-15-cf904ab46dd7>", line 1
    for i in range(0,99):
                        ^
SyntaxError: unexpected EOF while parsing
```

The “SyntaxError: unexpected EOF while parsing” error occurs when the end of your source code is reached before all code is executed because of some structural mistake.

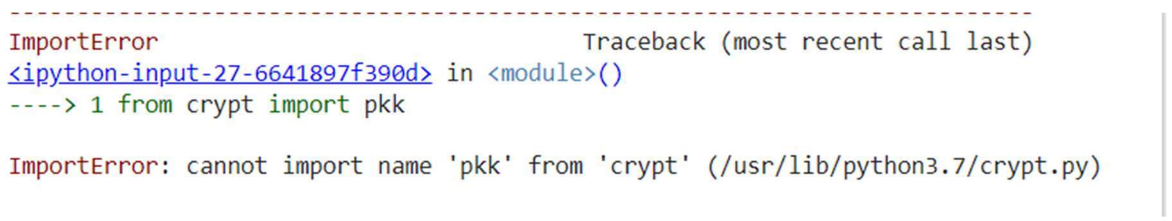
Keyerror :


```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-22-eac8b384a306> in <module>()
      3     'hi':2
      4 }
----> 5 mydict['bye']

KeyError: 'bye'
```

Here we are trying to print a key which is not present in dictionary.

ImportError:


```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-27-6641897f390d> in <module>()
----> 1 from crypt import pkk

ImportError: cannot import name 'pkk' from 'crypt' (/usr/lib/python3.7/crypt.py)
```

As we know there is no such thing as pkk in crypt , so it cannot be imported and thatswhy it shows ImportError.

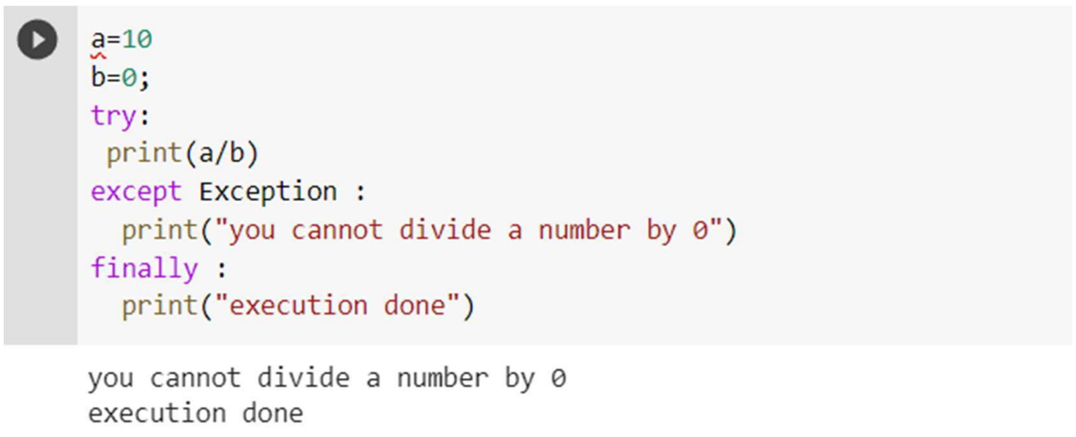
Handling exceptions in python :

We can handle these by using some statements such as – 1)try

2)throw

3)catch

Let's see one example –


```
a=10
b=0;
try:
 print(a/b)
except Exception :
  print("you cannot divide a number by 0")
finally :
  print("execution done")
```

```
you cannot divide a number by 0
execution done
```

As we know we cannot divide a number zero as the result is infinity so the condition written under the ” try” block will throw an error as we know if we will get any runtime error then execution will stop but in this case, execution will continue and because “try “condition failed then except condition will be executed and “finally” will always be executed at last

so it is clear that at first “try “ would be checked and if it fails then except will be executed. one try statement can have more than one except statement for handling different exceptions.

Syntax :

Try :

#condition

Except ImportError:

#condition

Except ModuleNotFoundError :

Except NameError :

#condition

Finally :

#condition

If the “try” condition fails and the exception is NameError

Then “Except NameError “ will be executed and then “finally” will be executed.

If the “try” condition fails and the exception is ImportError

Then “Except ImportError “ will be executed and then “finally” will be executed.