Execution/Running a program

Generally, we developed the programs in high level language like C, C++, Java etc ,which cannot understand by the computer . It can understand only low level language. So, the program written language in high level to be converted into low level language to make it understandable for the computer. This conversion is performed either using Compiler or interpreter.

Low level language(which is not understandable for the human beings- assembly language(coded language)-assembler and binary language(machine language)

Debugging

Debugging means the process of finding errors, finding reasons of errors and techniques of their fixation.

An error, also known as a bug, is a programming code that prevents a program from its successful interpretation.

Errors are of three types –

- Compile Time Error
- Run Time Error
- Logical Error

Debugging

Compile time error :

These errors are basically of 2 types –

Syntax Error :Violation of formal rules of a programming language results in syntax error.

For ex-

```
a=5
if a>10:
SyntaxError: ‘:’ syntax error
```

Semantics Error: Semantics refers to the set of rules which sets the meaning of statements. A meaningless statement results in semantics error.

For ex-

```
x * y = z
```

Debugging

Logical Error

If a program is not showing any compile time error or run time error but not producing desired output, it may be possible that program is having a logical error. These are not detected by the complier or interpreter. These are hard to detect.

Some example-

- Use a variable without an initial value.
- Provide wrong parameters to a function
- Use of wrong operator in place of correct operator required for operation

X=a+b (here – was required in place of + as per requirement

Run time Error

These errors are generated during a program execution due to resource limitation.

Python is having provision of checkpoints to handle these errors.

For ex-

```
a=10
b=int(input("enter a number"))
c=a/b
```

Value of b to be entered at run time and user may enter 0 at run time,that may cause run time error, because any number can't be divided by 0

Debugging

Run time Error

In Python, try and except clauses are used to handle an exception/runtime error which is known as exception handling

```
try:
#       code with probability of exception will
        be written here.
        a=10
        b=int(input(“enter a number”))
        c=a/b
except:
        #code to handle exception will be written here.
        print(“divide by zero erro”)
```

Debugging

Run time errors- Available exception in python

Exception Name	Description
IOError	This exception generates due to problem in input or output.
NameError	This exception generates due to unavailability of an identifier.
IndexError	This exception generates when subscript of a sequence is out of range.
ImportError	This exception generates due to failing of import statement.
TypeError	This exception generates due to wrong type used with an operator or a function.
ValueError	This exception generates due to wrong argument passed to a function.
ZeroDivisionError	This exception generates when divisor comes to zero.
OverFlowError	This exception generates when result of a mathematical calculation exceeds the limit.
KeyError	This exception generates due to non-availability of key in mapping of dictionary.
EOFError	This exception generates when end-of-file condition comes without reading input of a built in function.