**IT1140 PYTHON PROGRAMMING**

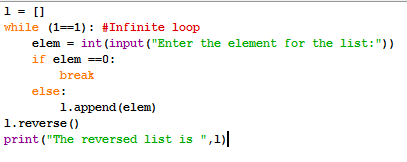
**B**

**CYCLE TEST 1**

**MAX MARKS: 25 DATE: 02.08.2016**

**PART – A (ANSWER ANY THREE) 3\*4 = 12 marks**

1. **Get input numbers from the user .The user will enter 0 as a sentinel value to indicate that no further values will be provided. Reverse the list in place.**



1. **Explain the output of the below program.**

**l = [5,7,8]**

**try:**

**l+"hello"**

**except TypeError:**

**print("This operation is not supported")**

**else:**

**print("No exception")**

**finally:**

**print("Reached finally")**

**print("End of program")**

The flow would be except block – finally block and then program resumes execution.

**Output:**

**This operation is not supported**

**Reached finally**

**End of program**

1. **Enumerate the functionalities of a PVM.**

–Iterates through byte code

–Executes instruction one by one

–No build or make steps

–Byte code is not machine code

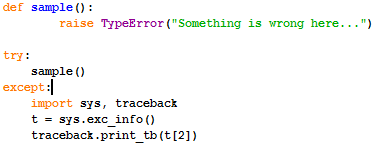
–Python representation

1. **Given the function definition**

**def sample():**

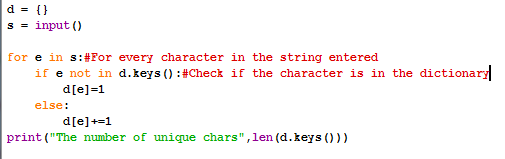
**raise TypeError("Something is wrong here...")**

**Write a try catch block and print the stack trace of the raised exception.**



**PART – B 1\*13 = 13 marks**

1. Create a program that determines and displays the number of unique characters in a string entered by the user. For example, “hello, world!” has 10 unique characters while “zzz” has only one unique character. Use a dictionary or set to solve this problem.Assume the string does not have spaces and all characters in lower case.**(13 marks)**



**(OR)**

1. Create a program that reads words from the user until the user enters 0. After the user enters 0 the program should display each word entered by the user exactly once. The words should be displayed in the same order that they were entered. Use lists. For example, if the user enters:

first

second

first

third

second

then your program should display:

first

second

third **(13 marks)**

