



Welcome to CSE1002

Venue: AB1-404

Semester: Winter 2017-18

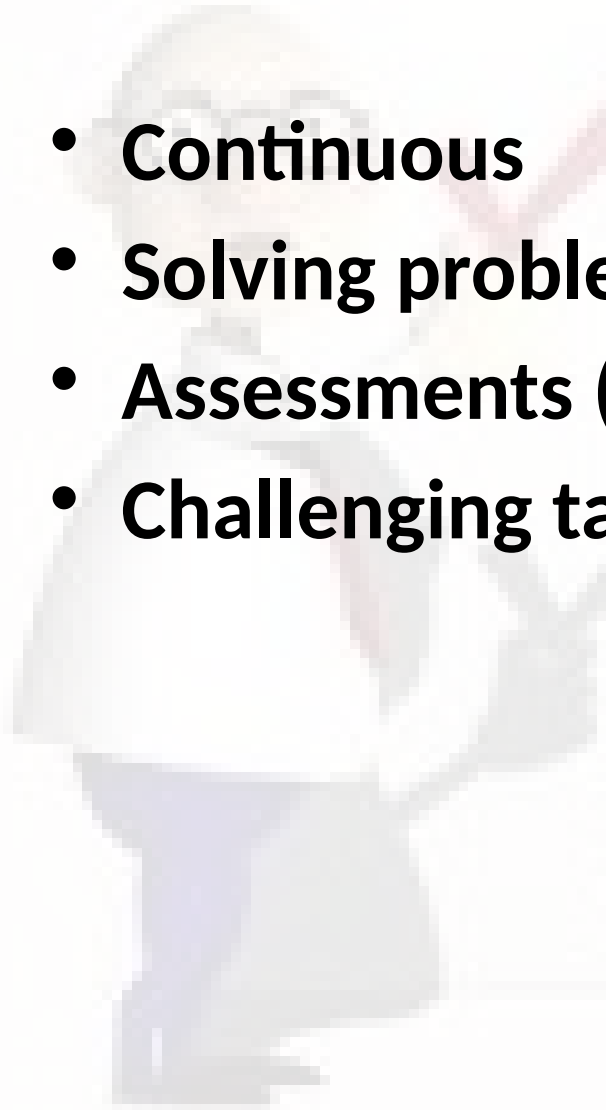
Slot: L3+L4, L21+L22, L25+L26

About the Course

- **A lab only course for problem solving and coding skill development**
 - **What is Skill? How does it differ from knowledge and information?**
 - **Painting, carpentry, welding, singing, dancing, swimming etc are skills**
 - **How to gain it?**
 - **Can be obtained only through continuous practice**

Evaluation

- **Continuous**
- **Solving problems in class - 40%**
- **Assessments (Four) - 20%**
- **Challenging task (TWO) - 40%**



OUTSTANDING

Excellent

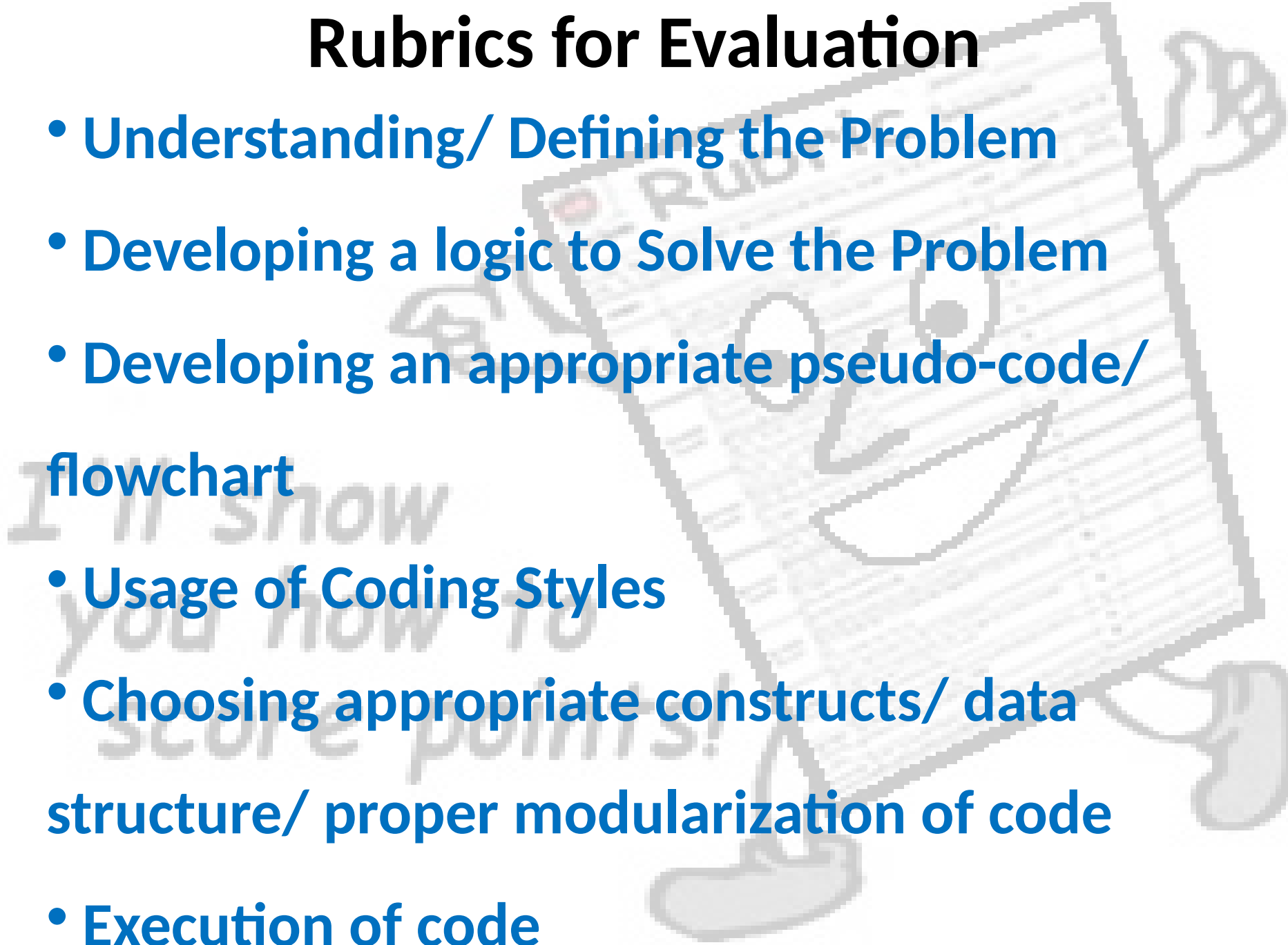
Very Good

Average

Below Average

Rubrics for Evaluation

- Understanding/ Defining the Problem
- Developing a logic to Solve the Problem
- Developing an appropriate pseudo-code/
flowchart
- Usage of Coding Styles
- Choosing appropriate constructs/ data
structure/ proper modularization of code
- Execution of code



About the Me

Name: Tulasi Prasad Sariki

Areas of Interest : NLP, Data Science, ML.

Cabin: AB1-604 Cabin No-13

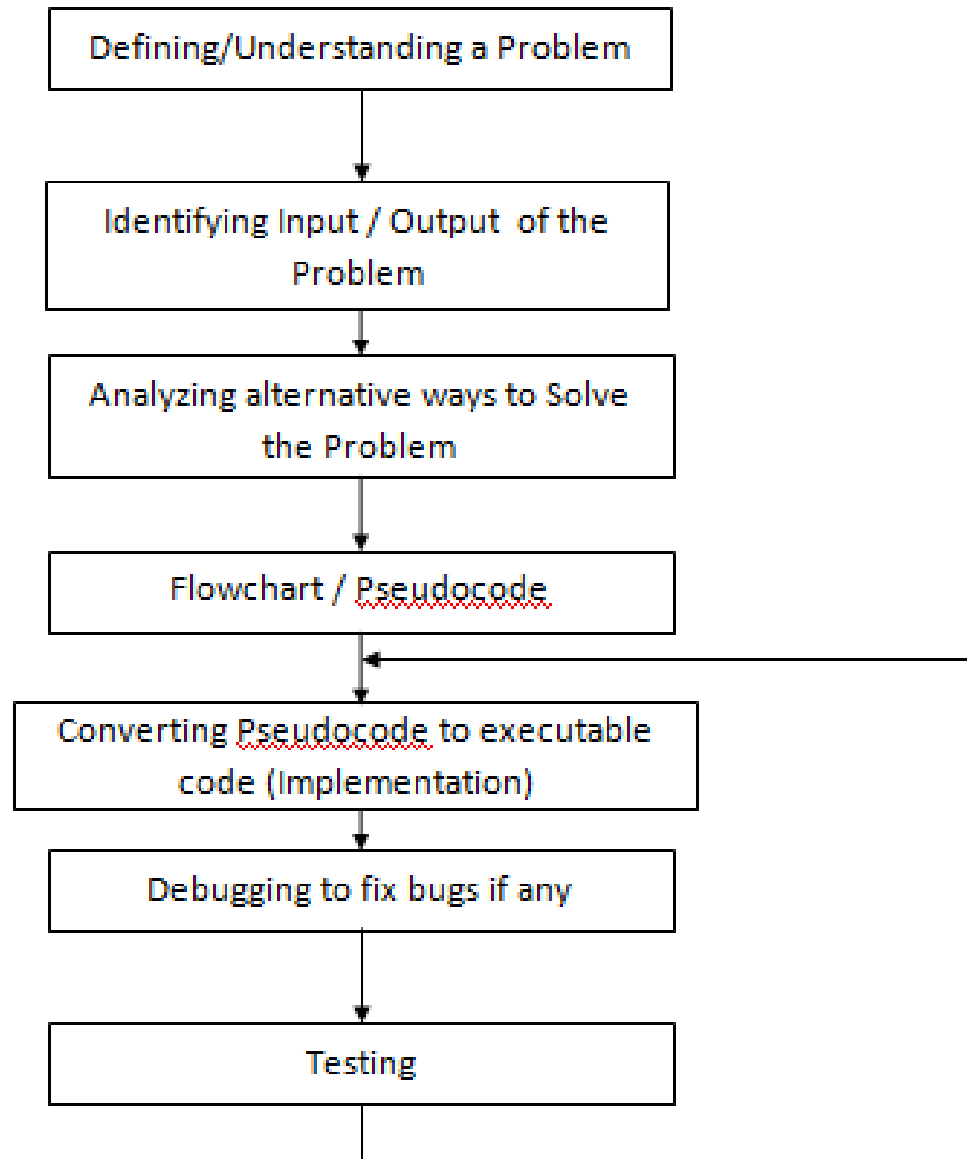
Phone Number: 8675724402

Email : tulasiprasad.sariki@vit.ac.in

website: www.learnersdesk.weebly.com



Problem Solving Steps – A Recap





Why did We Learn Python?

- Easy to learn
- Language with simple rules
- Good for beginners
- Code is readable
- Less development time
- No memory management
- Great support for building web apps
- Dynamic language and no type checking



Limitations of Python

Python is not a good choice for:

- **Memory intensive and computation intensive tasks**
- **Embedded Systems where processor has limited capacity**
- **For graphic intensive 3D game that takes up a lot of CPU**
- **Applications that demand concurrency and parallelism**
- **Developing mobile apps**
- **Design restrictions**
- **Interpreted language and is slow compared to C/C++ or Java**

The logo for C/C++ programming languages, featuring the text 'C/C++' in a bold, blue, sans-serif font. The background of the logo is a light blue square with a blurred image of a hand typing on a keyboard.

Why to learn more languages?

- Similar to why a carpenter has more than just a hammer in his/her toolbox
- Every programming language has its positive and negative points
- One language cannot do everything
- That is why there are many languages; some are fantastic for some things
 - Eg: C/C++ is typically the benchmark for speed and memory usage, and some languages provide strengths elsewhere (Eg: Python is very easy to pick up)

Transiting from Python to C/C++

- Will not be so hard
- There are quite a few syntax differences between the two languages
- Only way to learn a new programming language is by writing programs in it

– Dennis Ritchie

History of C

- Born at AT & T Bell Laboratory of USA in 1972
- Many of C's principles and ideas were derived from the earlier language B
- Ken Thompson was the developer of B Language
- C was written by [Dennis Ritchie](#)

http://www.nytimes.com/2011/10/14/technology/dennis-ritchie-programming-trailblazer-dies-at-70.html?_r=0

- C language was created for a specific purpose i.e designing the UNIX operating system (which is currently base of many UNIX based OS)
- Quickly spread beyond Bell Labs in the late 70's because of its [strong features](#)

About Dennis Ritchie

- Born September 9, 1941
- Known for ALTRAN, B, BCPL, C, Multics, Unix
- Won Turing Award in 1983
- Developed C language which is widely used developing, operating systems, compiler, and embedded system development, Assemblers, Text editors, Print Spoolers, Network drivers databases etc and its influence is seen in most modern programming languages
- Died on October 12, 2011

Features of C language

- Portability - C Programs can run on any compiler with little or no modification
- Low level features: C provides low level features and is closely related to lower level **assembly Languages**
- Modular programming - software design technique that increases the extent to which software is composed of separate parts, called **modules**
- Has many successor languages which are designed to look like C, e.g., C++, C#, Objective-C, Java, JavaScript, PHP and Perl.

C is a structured programming language

- Divides the large problem in to smaller modules called functions or procedures
- Each function or module handles the particular task and the collection of all the functions is called a program, which solves the large problem
- Easier to modify and debug

Modularity in Technology and Management

- Product systems are deemed “modular”
- They can be decomposed into a number of components that may be mixed and matched in a variety of configurations
- Components are able to connect, interact, or exchange resources
- Plugs and plug points are independent, may be manufactured even by different companies

Difference between Python and C

Low level →

- **C programs – Compiled**

High level →

- **Python programs – Interpreted**



1010010110111010
1001110110000111
0001110010110001
1011010110111010
1001110010011101

```
sale_price = 1.00  
if (sale_price > 2) {  
    discount = 0.1  
}  
else {  
    discount = 0.05  
}
```

Compiler

Takes **entire** program as input and generate a output file with object code

Errors are displayed after **entire program** is checked

Interpreter

Takes instruction by instruction as input and gives an output. But does not generate a file

Errors are displayed for **every instruction** interpreted (if any)

Variable Declaration in C

- In C, it is mandatory to do variable declaration
- We say variable's **type**, whether it is an integer (**int**), floating-point number (**float**), character (**char**) etc
- Syntax is type of variable, white space, name of variable semicolon
- Eg: `int number;`



Container

White spaces and Indentation

Block 1

- No problem of difference between white space and tab in C (Happy!)

Block 2

- Block of code in C need not be intended as in Python

- In C, Curly braces are used for giving a block of code

Block 3

Eg: Block of code in 'C'

```
{  
    Block 2, continuation
```

```
-----
```

```
}  
Block 1, continuation
```