**Basics Of C Programming**

D:\zigbee_street_light\logo.jpg

Block A - 7th Floor,Safal Profitaire,

Corporate Road, Prahladnagar,

Ahmedabad-380 015,

Gujarat, India

email: [info@volansys.com](mailto:info@volansys.com)

**Contents**

[Objective](#gjdgxs)

[Prerequisite Knowledge](#1fob9te)

[Total Duration](#3znysh7)

[Completion Criteria & Rating](#2et92p0)

[Evaluation criteria](#tyjcwt)

[HW/SW Required](#3dy6vkm)

[Instructions](#1t3h5sf)

[List of Exercises](#4d34og8)

[Exercise Set#1](#kix.6vjzcb3l4qyb)

[Exercise Set#2](#2s8eyo1)

[Exercise Set#3](#17dp8vu)

[Exercise Set#4](#kix.ib7hmd7k43d)

[Exercise Set#5](#3rdcrjn)

[Exercise Set#6](#26in1rg)

[Exercise Set#7](#lnxbz9)

[Exercise Set#8](#2jxsxqh)

[Exercise Set#9](#kix.aa2w4av2svmb)

[Exercise Set#10](#kix.jjqvj56z52e1)

[Exercise Set#11](#3j2qqm3)

[MEGA Exercise](#1ci93xb)

**History**

| Version | Author | Date | Comments |
| --- | --- | --- | --- |
| 0.1 | HK | 18-Mar-2015 | First version |
| 0.2 | Amit Solanki | 15-Apr-2015 | Initial draft |
| 1.0 | Amit Solanki | 20-Jul-2015 | First version |
| 2.0 | Ashish Agraval | 01-June-2017 | Included doxygen, squeezed this basic training time in order to introduce advanced C training |
| 2.1 | Rajvi Kamdar | 07-July-2017 | Dynamic problem allocation and introduced mega exercise as a set of multiple dynamically allocated problems. |
| 2.2 | Rajvi Kamdar | 22-May-2020 | Modified mega exercise. |

# Objective

The objective of this training is to get thorough understanding of C programming language. At the end of the training the trainee should be proficient with following concepts of C programming:

* Coding guidelines
* Writing, compiling, and debugging C programs
* Variables, datatypes and operators
* Typedef, type casting and storage class
* Control Flow
* Function and Scope Rules
* Basic Pointers and Arrays
* Structure, Unions , Enumerations and bitfields
* Function Pointers
* Input/Output
* Static & Dynamic Libraries

At the end of training, trainee should be independently able to develop software using C programming language.

Apart from technical aspect, trainee should also learn how to do write a Software Requirement Document (SRD) and Software Design Document (SDD).

# Prerequisite Knowledge

* Linux Essential training, specifically basic linux commands and Vim editor usage.
* Basic understanding of the principles of programming. Prior knowledge of C is neither expected nor desired.

# Total Duration

20 Days

# Completion Criteria & Rating

* Trainee should score above the minimum points to complete the training, or else he/she will not qualify for next training:
  + Total Points: 150
  + Minimum Point: (90)
* Points will be assigned based on the mentor’s best assessment of your understanding of the C programming language. Any two Artifacts out of the exercises will be reviewed by the mentor, each carrying 4 points. 2 Points are dependent on answers to the mentor’s questions. Details of scoring points are described in each exercise.

| Rating | Meaning | Criteria |
| --- | --- | --- |
| A | Above Expectation | > 135 |
| B | As Expected | > 105 |
| C | Below Expectation | > 90 |
| D | Not Acceptable | < 90 |

# Evaluation criteria

* The exercises for each day should be submitted on the same day. 1 Point will be deducted for any delay. Evaluation will be done only when exercises are completed.
* Trainee should be able to explain the logic used in the program.
* Follow coding conventions as defined in the [C Coding Standard Guidelines](https://docs.google.com/document/d/1237QX0u6UEtZlYnvTpzV2vurjxpredSr9z6b4lwDVIk).
* Cheating will not be tolerated. 5 Point will be deducted if you are caught copying code from internet or fellow trainees.
* Bonus 2 points will be awarded for completing a day’s exercise beforehand.

# HW/SW Required

* A standard PC running Ubuntu 14.02 LTS OS or higher.

# Instructions

* Use Vim editor or Eclipse CDT for programming
* Use GCC compiler with ‘-Wall’ option. There should not be any compilation warnings.
* Write each exercise in separate source (.c) file.
* Write problem statement in the beginning of each file.
* Use comments and follow Coding style.

# List of Exercises

## Exercise Set#1

Coding Guidelines and hands-on with Doxygen tool

| Exercise Details |  |
| --- | --- |
| Duration | 1 Day |
| Reading Material | 1. C Coding Standard Guidelines.docx 2. Download “**doxygen-1.8.13.linux.bin.tar.gz”** from *Part-1 material>Exercise 1* and Install it 3. Download “[doxygen\_manual-1.8.13.pdf.zip](http://ftp.stack.nl/pub/users/dimitri/doxygen_manual-1.8.13.pdf.zip)” from *Part-1 material>Exercise 1*. Refer Section 3.1 to 3.4 , 4.1.1 |
| Audio/Video | NA |
| Outcome/  Artifacts | Understand coding guidelines properly. If required, discuss the same with your mentor  Write hello world program. Use Doxygen style commenting and generate “.html” document  Note: Follow Doxygen style commenting method for all the programs in this training and generate html document |
| weightage | 10 Points |

## Exercise Set#2

Writing, compiling, and debugging C programs

| Exercise Details |  |
| --- | --- |
| Duration | 1 Day |
| Reading Material | 1. <https://www.tutorialspoint.com/cprogramming/c_program_structure.htm> 2. <https://www.hackerearth.com/practice/notes/what-happens-when-a-c-program-runs/> 3. MIT6\_087IAP10\_lab01.pdf 4. C/C++ Development User Guide in Eclipse CDT Help 5. Compilation Stages.pdf |
| Audio/Video | 1. C Programming - Introducing C.mp4 2. C Programming - First Steps.mp4 |
| Outcome/  Artifacts | 1. Trainee should be able to write, compile, run and debug a simple C program in Vim or Eclipse. (6 Points) 2. Explain Compilation Stages (2 Points) 3. Explain Different segments of the executable file (2 Points) |
| weightage | 10 Points |

## Exercise Set#3

Variables, datatypes and operators.

| Exercise Details |  |
| --- | --- |
| Duration | 0.5 Day |
| Reading Material | <https://www.tutorialspoint.com/cprogramming/c_data_types.htm>  <https://www.tutorialspoint.com/cprogramming/c_variables.htm>  <https://www.tutorialspoint.com/cprogramming/c_constants.htm>  <https://www.tutorialspoint.com/cprogramming/c_operators.htm> |
| Audio/Video |  |
| Outcome/  Artifacts | Understand various data types, operators and variable usage properly. |
| weightage | 5 Points |

## Exercise Set#4

Typedef, type casting and storage class.

| Exercise Details |  |
| --- | --- |
| Duration | 0.5 Day |
| Reading Material | <https://www.tutorialspoint.com/cprogramming/c_typedef.htm>  <https://www.tutorialspoint.com/cprogramming/c_type_casting.htm>  <https://www.tutorialspoint.com/cprogramming/c_storage_classes.htm>  <https://www.tutorialspoint.com/objective_c/objective_c_preprocessors.htm> |
| Audio/Video |  |
| Outcome/  Artifacts | Understand how & when to use typedef, and type casting. Also get aware about various storage class. Your mentor will ask questions to verify your understanding. |
| weightage | 5 Points |

## Exercise Set#5

Control flow.

| Exercise Details |  |
| --- | --- |
| Duration | 1 Day |
| Reading Material | Decision Making:  <https://www.tutorialspoint.com/cprogramming/c_decision_making.htm>  <https://www.tutorialspoint.com/cprogramming/if_statement_in_c.htm>  <https://www.tutorialspoint.com/cprogramming/if_else_statement_in_c.htm>  <https://www.tutorialspoint.com/cprogramming/nested_if_statements_in_c.htm>  <https://www.tutorialspoint.com/cprogramming/switch_statement_in_c.htm>  <https://www.tutorialspoint.com/cprogramming/nested_switch_statements_in_c.htm>  Looping:  <https://www.tutorialspoint.com/cprogramming/c_loops.htm>  <https://www.tutorialspoint.com/cprogramming/c_for_loop.htm>  <https://www.tutorialspoint.com/cprogramming/c_while_loop.htm>  <https://www.tutorialspoint.com/cprogramming/c_do_while_loop.htm>  <https://www.tutorialspoint.com/cprogramming/c_nested_loops.htm>  <https://www.tutorialspoint.com/cprogramming/c_break_statement.htm>  <https://www.tutorialspoint.com/cprogramming/c_continue_statement.htm>  <https://www.tutorialspoint.com/cprogramming/c_goto_statement.htm> |
| Audio/Video |  |
| Outcome/  Artifacts | Understand how and when to use loops, if-else-else if statements. Practice and experiment examples as given in the above given Reading Material. |
| weightage | 10 Points |

## Exercise Set#6

Functions and scope rules.

| Exercise Details |  |
| --- | --- |
| Duration | 1 Day |
| Reading Material | 1. <https://www.tutorialspoint.com/cprogramming/c_functions.htm> 2. <https://www.tutorialspoint.com/cprogramming/c_function_call_by_value.htm> 3. <https://www.tutorialspoint.com/cprogramming/c_function_call_by_reference.htm> 4. <https://www.tutorialspoint.com/cprogramming/c_scope_rules.htm> 5. <https://www.tutorialspoint.com/cprogramming/c_recursion.htm> 6. <https://www.tutorialspoint.com/cprogramming/c_variable_arguments.htm> |
| Audio/Video |  |
| Outcome/  Artifacts | Understand function calls by value and call by reference, scope of variables and recursive use of functions. Experiment with the examples given from the reading material and understand the concept. |
| weightage | 10 Points |

## Exercise Set#7

Basic Pointers and Array.

| Exercise Details |  |
| --- | --- |
| Duration | 2 Day |
| Reading Material | 1. <https://www.tutorialspoint.com/cprogramming/c_arrays.htm> 2. <https://www.tutorialspoint.com/cprogramming/c_multi_dimensional_arrays.htm> 3. <https://www.tutorialspoint.com/cprogramming/c_passing_arrays_to_functions.htm> 4. <https://www.tutorialspoint.com/cprogramming/c_return_arrays_from_function.htm> 5. <https://www.tutorialspoint.com/cprogramming/c_pointer_to_an_array.htm> 6. <https://www.tutorialspoint.com/cprogramming/c_pointers.htm> 7. <https://www.tutorialspoint.com/cprogramming/c_pointer_arithmetic.htm> 8. <https://www.tutorialspoint.com/cprogramming/c_passing_pointers_to_functions.htm> 9. <https://www.tutorialspoint.com/cprogramming/c_return_pointer_from_functions.htm> 10. <https://www.tutorialspoint.com/cprogramming/c_strings.htm> |
| Audio/Video |  |
| Outcome/ Artifacts | Understand basic pointers and arrays, how to pass pointers to functions, when to use arrays and pointers and experiment with the examples provided to you in the reading material. |
| weightage | 20 Points |

## Exercise Set#8

Structure, Unions & Enumeration

| Exercise Details |  |
| --- | --- |
| Duration | 1 Day |
| Reading Material | 1. <https://www.tutorialspoint.com/cprogramming/c_structures.htm> 2. <https://www.tutorialspoint.com/cprogramming/c_unions.htm> 3. <https://www.programiz.com/c-programming/c-enumeration> 4. <https://www.tutorialspoint.com/cprogramming/c_bit_fields.htm> |
| Audio/Video |  |
| Outcome/  Artifacts | Understand structures, union and enums. Difference between all the three, where they should be used, under what case each one is used based on optimum memory utilization. |
| eightage | 10 Points |

## Exercise Set#9

Function Pointers

| Exercise Details |  |
| --- | --- |
| Duration | 1 Day |
| Reading Material | 1. Refer following topic from : <http://www.newty.de/fpt/fpt.html>  * Introduction to Function Pointers * The Syntax of C and C++ Function Pointers * How to Implement Callbacks in C and C++ ? * Functors to encapsulate C and C++ Function Pointers   Note: Skip information/sample example for C++ |
| Audio/Video |  |
| Outcome/  Artifacts | Understand function pointers and experiment based on the reading material mentioned. |
| weightage | 10 Points |

## Exercise Set#10

Input and output.

| Exercise Details |  |
| --- | --- |
| Duration | 1 Day |
| Reading Material | 1. <https://www.tutorialspoint.com/cprogramming/c_input_output.htm> 2. <https://www.tutorialspoint.com/cprogramming/c_file_io.htm> |
| Audio/Video |  |
| Outcome/  Artifacts | Understand input output system, get to know more about file operations, experiment examples given in the reading material mentioned. |
| weightage | 10 Points |

## 

## Exercise Set#11

Static and Dynamic Libraries

| Exercise Details |  |
| --- | --- |
| Duration | 1 Day |
| Reading Material | 1. MIT6\_087IAP10\_lec09.pdf (Pages 6-27) 2. Static and Dynamic Link Libraries.pdf |
| Audio/Video |  |
| Outcome/  Artifacts | Understand static and dynamic libraries, experiment with examples given in reading material. |
| weightage | 10 Points |

## MEGA Exercise

| Exercise Details |  |
| --- | --- |
| Duration | 10 Days |
| Reading Material | Develop a custom malloc and free functions which will imitate the actual malloc and free functions. Understand how malloc and free works (the metadata structure), then implement the same. Function : myMalloc() and myFree()    **AND**  Ask your mentor for mega exercise.  Note: Here are some reference links for the mentor which can be used to select mega exercise. Mentor can refer any other links as well/define the exercise on its own.   * [www.codechef.com](http://www.codechef.com) * [https://www.hackerrank.com](https://www.hackerrank.com/CProgramming?h_r=internal-search&hr_r=1) * <https://www.udemy.com/>   Mega Exercise should consist of all the trainings to be done based on all of the above given topics and the problem allocation should be completely dynamic.  Based on the overall evaluation of the candidate from all above exercise, mentor shall select appropriate level of mega exercise - Beginner/easy/medium/hard/challenge. Following number of points will be deducted from your points based on the exercise level selected.  Beginner : 20 points deducted  Easy : 15 points deducted  Medium : 10 points deducted  Hard/challenge : 0 points deducted.  i.e. If trainee got 135 points out of total 150 points and he/she has appeared for easy mega exercise level, 15 points will be deducted. So trainee’s point will be 135-15=120 points |
| Audio/Video |  |
| Outcome/  Artifacts | The project includes:   1. Writing a Software Requirement Document 2. Writing a Software Design Document 3. Writing a Test case Document 4. Development 5. Testing |
| weightage | 40 Points |