**C**

1. **Module-00: Orientation and Welcome**
2. **welcome video**
3. if you survive this course you will be significant a software developer afer 1 year.
4. You must be sacrifice viral,hot,love,emotions etc.otherwise your career will be damaged.
5. This course must be top priority your daily life.
6. If you want to good anything,you must be do hard work and sacrifice life,and it’s not easy.
7. You need to make sacrifice.
8. it’s would not be easy.
9. Anybody you can’t do becoming a programmer.
10. **Orientation Welcome**
11. why in this student fallback down in this course? Just one reason,they are not understand and do not take support in this courese mentor.
12. **Course curriculum and course alignment**
13. math course
14. introduction c with lab.bundling course.
15. Introduction c++.stand alone course
16. basic data structure with lab.bundling course.
17. Basic algorithms with lab.bundling course.
18. Advanced problem solving or software engineering
19. **build a routine and adjust with routine**
20. sacrifice unwanted time
21. list of compromised and uncompromised things
22. make a time line
23. student time line
24. job holder time line
25. bekar time line
26. housewife time line
27. **a little overview of exans assignment and deadlines**
28. don’t avoid any learning topics for exam or assingment.because goal is not pass exam goal is job market future. So you don’t konw what’s will be need to for future job carry.
29. Don’t salaki for exam or assingment.because if you salaki with exam or assignment you must be upset for future.
30. Don’t use shrotcut.no shortcut for job.
31. Somoyer ekpor osomoyer dos por.
32. **Best way to get support from this course**
33. if you faced a problem directly don’t go support you selfed try it few times.
34. Error search to google.
35. Go group support scroll then post your scroll.
36. Go to support session at a times-11am,4pm,11pm
37. you can email support – support@phitron.io
38. you can call support - 01322810881 , 01322810876 (Available : 10AM - 5PM)
39. support for google meet use.
40. Advice – don’t take lack of helping mind. We also don’t take lack or helping mind.so,something happened good.
41. **How to get best outcome from this course**
42. sacrifice – you need to sacrifice just for 1 year.
43. Patience – spend time with patience.
44. Dynamic routine – make a dynamic routine.
45. If you take continue – something would be good happened.
46. Resource link for setup- https://docs.google.com/document/d/1LKz1Ilmae1O8fR-SKBbGttZ7JVDDIanccPI6MwFPgiI/edit
47. **how to join support session**
48. submit your problem into support forum.
49. **How to use google meet for support session.**
50. Screen share
51. chatting
52. speaker mute on or off
53. **website hard reload and cookie clear**
54. log out/login
55. ctrl+shift+r = hard reload
56. cache clear
57. group post
58. **install vs code,git bash, zoom**
59. vs code = for editor
60. git bash = git terminal
61. zoom = conceptual session & live support
62. **set up shortcut key and input and output**
63. code run key = ctrl+r
64. setting.json file add their code
65. **codeblocks setup**
66. editor with compilar
67. resource link for github batch 4- https://github.com/phitronio/Batch-4-Introduction-to-C-Programming
68. **Module-01: Basic Syntax, Variables and Data Types**
69. **introduction**
70. c invented by Dennis Ritchie since 1972
71. c is mother of programming language
72. c is easy
73. c almost low level language
74. in this chapter we will be learn: input,output,variable,data type,data type limitation,rules for variable names;
75. **basic structure of c programming.**
76. C programming structure like human body.
77. Header file = head of brain station
78. main body = brain surve their conneciton into body.
79. **first c program.**
80. Built-in function
81. header file
82. printf()
83. **understand how to print in c**
84. header part.
85. body part.
86. user defined function.
87. built in function.
88. function like robots.
89. Any language run from a main function
90. statement closed must be used (;) semicolon.
91. header file.
92. library file.
93. #include.
94. stdio=standard input and output.
95. every function has a return type.
96. **run first c program**
97. printf “Hello world”
98. spacial character: \,%
99. escape character: \n, \t, \\, %%,
100. c language comments: //comment\_here or /\*comment\_here\*/
101. **Variables and data types.**
102. Variable is like container;
103. data is a assets of variable;
104. data types: int, float, char, bool, double, long;
105. char data assign in a single quotation.
106. Declare = just varible declare but value not assign;
107. assign = after variable declare value assign into variable;
108. initialization = declare and assign at a time;
109. **Variables and data types in c.**
110. after decalring a variable stored temporary in memory.
111. Variable declare vs variable assign
112. But how many memory consume diferent data type?
113. Int data = 4 bytes
114. float data = 4 bytes
115. char data = 1 byte
116. long long int data = 8 bytes
117. double data = 8 bytes
118. 1g = 1024 mb
119. 1mb = 1024 kb
120. 1kb = 1024 byte
121. 1byte = 8bit
122. format specifier: %d = int; %f = float; %c = char; %lf = double; %lld = long long int; %s = string array;
123. %0.2f
124. **why we need to take input.**
125. For dynamic action
126. **How to take input?**
127. Scanf(“%d”,&a);
128. scanf(“%da”, &a);
129. print(“%d”,a);
130. **Data types limitations in c.**
131. 1 bit = 0 or 1
132. int a = 10 digits+
133. long long int a = 20 digits
134. float a = 6 digits
135. double a = 15 digits
136. **rules of naming variables.**
137. Variable name must start with a letter or underscore
138. variable must contain letter, digits or underscore
139. kywords can’t be used as variables
140. **Quiz**
141. **Module-02: Operators, Conditional Statements**
142. **introduction**
143. arithmetic operators
144. relational operators
145. conditional statements
146. control based on conditions
147. conditional statements in c programming
148. if..else,nested...if..else learning
149. **operators**
150. operators is like a symbol
151. **arithmetic operators**
152. addition(+)
153. multiplication(\*)
154. subtraction(-)
155. division(/)
156. remider(%)
157. one floating must have for complete decimal value
158. **relational operators**
159. equal (==)
160. not equal (!=)
161. greater than (>)
162. greater than or equal (>=)
163. less than (<)
164. less than or equal (<=)
165. **logical operators**
166. and (&&)
167. or (||)
168. not (!)
169. **conditional statement type**
170. condition means condition
171. if..else in c programming
172. conditional statement type-2
173. if..else ladder in c programming
174. conditional statement type
175. nested if..else in c programming
176. **nesed if..else.. process**
177. if one condition is true,then next all conditon will not be processed
178. **summary**
179. final motivate
180. **quiz**
181. **Module-03: Loop**
182. **Introduction**
183. introduction loop
184. where to use loop
185. **for..loop type-1**
186. for(*initialization,condition,increment/decrement*)
187. initialization just first time load.
188. Concept of looping process
189. for...loop type-1
190. for...loop type-2
191. for...loop type-2
192. sum from 1 to N
193. loop with conditions
194. break statement
195. continue statement
196. while and do..while loop
197. how to save yourself from infinity loop
198. **quiz**
199. **Module-04: Assignment 01**
200. assignment 01
201. **Module-05: Conditional Statements Recap**
202. **introduction**
203. https://snippet-generator.app/
204. vs code>settings>user snippets
205. $0
206. **welcome for you with conditions**
207. uses: if..else conditions
208. uses: >= operators
209. **multiples**
210. uses: %, ||, == operator
211. **char**
212. char use: ‘’ single quote
213. ASCII Value: %d
214. a=97,z=122
215. A=65, Z=90
216. diff=32
217. ‘0’=48,’9’=57
218. **first digit**
219. uses: / afetr % and minimum
220. **capital or small or digit**
221. 0=48,9=57
222. **max and min**
223. uses: >,<,&& operator
224. quiz
225. **Module-06: Loop Recap and Nested Loop**
226. **introduction**
227. vs code>extension>cph=competitive programming helper
228. chrome>competitive companion
229. **increment/decrement operator**
230. uses: ++, --;
231. words: post increment,pre-increment;
232. words: post decrement,pre-decrement;
233. **even numbers**
234. Given a number N. Print all even numbers between 1 and N inclusive in separate lines. If there are no even numbers print -1.
235. uses: condition,%,loop
236. **even, odd, positive and negetive**
237. uses: %,>,<,loop,condition
238. **fixed password (EOF)**
239. !EOF
240. **find maximum value**
241. uses: condition,loop
242. **max and min**
243. same max
244. **multiplication table**
245. uses: \*,loop
246. digits
247. nested for loop
248. nested for loop implementation
249. quiz
250. **Module-07: Introduction to Array**
251. **Introduction**
252. about array
253. why need to array?
254. How to save in memory and access it?
255. Array operation
256. **what is array?**
257. data\_type name[size]
258. **why we need array?**
259. Multiple data store
260. array memory addresed of sequentially her space 534-538
261. **accessing an array**
262. first index remember then access data type byte sequence.
263. **Array input and output**
264. input: arr[index]=value
265. output: arr[index]
266. **array initialization**
267. int arr[] = {values};
268. **reverse of array**
269. decreasing for loop
270. **sum of array**
271. using for loop summation
272. **Module-08: Assignment 02**
273. **Module-09: Array Operation**
274. **Introduction**
275. element insert in an array
276. remove an element into an array
277. array reverse
278. two pointers technique
279. array copy
280. **insert element in an array**
281. take a extra space
282. transfer array element being start last end;
283. **remove an element from an array**
284. take a remove index;
285. start loop from index and +1 add next index into current index
286. **reverse array(tow pointers technique)**
287. reverse mean array element swap first to last index;
288. need to a tmp value for first value holded;
289. **copy elements of two array**
290. just copying using and condition
291. need to increment using conditionaly
292. **retriving array length**
293. sizeof();
294. int length = sizeof(arr)/sizeof(int)
295. **Module-10: String**
296. **introduction**
297. what is string?
298. String input and output
299. get out of string length
300. string related problem
301. **what is String?**
302. Int array,float array = it’s a char array => string
303. why it’s a string and not called char array? Because it’s have a some power.
304. space also is a string
305. keyboard any key is a string
306. string behind the scene working as a array because it’s a char array.
307. String is character of array.
308. It’s have some super power.
309. **String Initialize and Print Super Power**
310. **initialize super power**
311. power of = “” doulbe quote;
312. char name[] = “Jagadish Chakma”;
313. \0 = for null value set
314. if you use “” double quote your don’t need to use \0 null vlaue;
315. for check actual size use: sizeof(name);
316. don’t need array init;
317. **print super power**
318. printf(“%s”,name);
319. dont’t need for loop;
320. if you not set size then automatically set ‘\0’ null of the end; if you set size.you manually add the ‘\0’ null;
321. **input super power**
322. scanf(“%s”,name);
323. don’t need: &,loop
324. but problem is: space is her end line;
325. **input super power with space**
326. fgets(name,size,stdin);
327. fgets take input alson enter key.it’s rember ok.
328. **string length**
329. ‘\0’=null;
330. using for loop:condition: namep[i] != ‘\0’;
331. using string.h: strlen();
332. **Rules**
333. scanf() when get a space he is imagine int’s new line or enter.
334. **Module-10.5: Practice Day**
335. [problem links](https://docs.google.com/document/d/1geZ2RYI0CXKm3O0Y5O9643MR8tlf82G_ozJexc06RrA/edit)
336. **Module-11: String Operation and Freequency Array**
337. **introduction**
338. string copy
339. compare two string
340. concatenation two string
341. counting array
342. **string copy**
343. using logic
344. using strcpy()
345. **string laxicographical comparison**
346. infinity loop declare
347. while(1),for(int I =0; ;i++)
348. condition is uses ASCII value compare
349. condition is uses nul \0 value
350. use strcmp() built in function
351. **string concatenation**
352. **counting or frequency array -1 using number**
353. count how many same value store in array
354. **counting or freequency array -2 using alphabet**
355. it’s amazing
356. just -97
357. **Module-13: Nested loop recape**
358. **Introduction**
359. nested loop
360. different nested loop pattern
361. one problem solve using nested loop
362. nested loop sorting algorithm
363. **nested loop pattern 1**
364. print \* given total time;
365. **nested loop pattern 2**
366. print tringle
367. **nested loop pattern 3**
368. print print tringle
369. **nested loop pattern 4**
370. print diamond
371. **problem solving**
372. sum of two values
373. **sorting alogrithom**
374. ascending or descending sorting an array
375. **Module-14: Function and Pointer**
376. **Introduction**
377. about function
378. different function
379. built in function
380. basic pointer
381. **useful built in function**
382. ceil()
383. floor()
384. round()
385. sqrt()
386. pow()
387. abs()
388. **scope**
389. %p,&a;
390. **pointer**
391. **Module-15: Function With Pointer**
392. **introduction**
393. about pointer
394. about call by value and call by reference
395. array relation with pointer
396. function working with array and string
397. array how to working call by refernce way in a function
398. **pointers in c**
399. refernce by reference
400. \*name used for pointers
401. **pass by value;**
402. pass a value;
403. **pass by reference**
404. pass address
405. **Module-17: Recurson Function**
406. **introduction**
407. call stack
408. recursion
409. print using recursion
410. recusion drive way of reverse
411. print array using recursion
412. find out length of string using recursion
413. **call stack**
414. call stack data structure
415. LIFO
416. wait
417. data structure like
418. call stack = function call like stack structure
419. **what is recursion**
420. self function calling called recursive function
421. **print from 1 to 5 using recursion**
422. base case
423. **print from 5 to 1 using recursion**
424. recursion is infinity
425. **print array using recursion**
426. just need 3 parameter
427. **length of string using recursion**
428. just awesome
429. **Module-18: 2D Array**
430. **declaration**
431. daty\_type array\_name[rows][columns]
432. **introduction**
433. 2D array
434. 2D array input and output
435. matrix and some matrix type
436. profes types using 2D array
437. **what is 2D array?**
438. 2D array like matrix
439. row\*column = [count\_array][size]
440. **2D array input and output**
441. row looping then column looping
442. **types of matrix**
443. row matirx
444. column matrix
445. zero matrix
446. diagonal matrix
447. scaler matrix
448. unit matrix
449. **Module-18.5: Practice Day**
450. absolute number print
451. stdlib.h
452. abs();
453. **Module-19: Recursion Recape**
454. **introduction**
455. mirror or inverted 2D array value;