**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;
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. operators
144. conditional statements
145. **arithmetic operators**
146. addition(+)
147. multiplication(\*)
148. subtraction(-)
149. division(/)
150. remider(%)
151. one floating must have for complete value
152. **relational operators**
153. equal (==)
154. not equal (!=)
155. greater than (>)
156. greater than or equal (>=)
157. less than (<)
158. less than or equal (<=)
159. **logical operators**
160. and (&&)
161. or (||)
162. not (!)
163. **conditional statement type**
164. condition means condition
165. if..else in c programming
166. conditional statement type-2
167. if..else ladder in c programming
168. conditional statement type
169. nested if..else in c programming
170. summary
171. quiz
172. **Module-03: Loop**
173. **Introduction**
174. for..loop type-1
175. for...loop type-1
176. for...loop type-2
177. for...loop type-2
178. sum from 1 to N
179. loop with conditions
180. break statement
181. continue statement
182. while and do..while loop
183. how to save yourself from infinity loop
184. quiz
185. **Module-04: Assignment 01**
186. assignment 01
187. **Module-05: Conditional Statements Recap**
188. **introduction**
189. https://snippet-generator.app/
190. vs code>settings>user snippets
191. $0
192. **welcome for you with conditions**
193. uses: if..else conditions
194. uses: >= operators
195. **multiples**
196. uses: %, ||, == operator
197. **char**
198. char use: ‘’ single quote
199. ASCII Value: %d
200. a=97,z=122
201. A=65, Z=90
202. diff=32
203. ‘0’=48,’9’=57
204. **first digit**
205. uses: / afetr % and minimum
206. **capital or small or digit**
207. 0=48,9=57
208. **max and min**
209. uses: >,<,&& operator
210. quiz
211. **Module-06: Loop Recap and Nested Loop**
212. **introduction**
213. vs code>extension>cph=competitive programming helper
214. chrome>competitive companion
215. **increment/decrement operator**
216. uses: ++, --;
217. words: post increment,pre-increment;
218. words: post decrement,pre-decrement;
219. **even numbers**
220. Given a number N. Print all even numbers between 1 and N inclusive in separate lines. If there are no even numbers print -1.
221. uses: condition,%,loop
222. **even, odd, positive and negetive**
223. uses: %,>,<,loop,condition
224. **fixed password (EOF)**
225. !EOF
226. **find maximum value**
227. uses: condition,loop
228. **max and min**
229. same max
230. **multiplication table**
231. uses: \*,loop
232. digits
233. nested for loop
234. nested for loop implementation
235. quiz
236. **Module-07: Introduction to Array**
237. **Introduction**
238. about array
239. why need to array?
240. How to save in memory and access it?
241. Array operation
242. **what is array?**
243. data\_type name[size]
244. **why we need array?**
245. Multiple data store
246. array memory addresed of sequentially her space 534-538
247. **accessing an array**
248. first index remember then access data type byte sequence.
249. **Array input and output**
250. input: arr[index]=value
251. output: arr[index]
252. **array initialization**
253. int arr[] = {values};
254. **reverse of array**
255. decreasing for loop
256. **sum of array**
257. using for loop summation
258. **Module-08: Assignment 02**
259. **Module-09: Array Operation**
260. **Introduction**
261. element insert in an array
262. remove an element into an array
263. array reverse
264. two pointers technique
265. array copy
266. **insert element in an array**
267. take a extra space
268. transfer array element being start last end;
269. **remove an element from an array**
270. take a remove index;
271. start loop from index and +1 add next index into current index
272. **reverse array(tow pointers technique)**
273. reverse mean array element swap first to last index;
274. need to a tmp value for first value holded;
275. **copy elements of two array**
276. just copying using and condition
277. need to increment using conditionaly
278. **retriving array length**
279. sizeof();
280. int length = sizeof(arr)/sizeof(int)
281. **Module-10: String**
282. **introduction**
283. what is string?
284. String input and output
285. get out of string length
286. string related problem
287. **what is String?**
288. Int array,float array = it’s a char array => string
289. why it’s a string and not called char array? Because it’s have a some power.
290. space also is a string
291. keyboard any key is a string
292. string behind the scene working as a array because it’s a char array.
293. String is character of array.
294. It’s have some super power.
295. **String Initialize and Print Super Power**
296. **initialize super power**
297. power of = “” doulbe quote;
298. char name[] = “Jagadish Chakma”;
299. \0 = for null value set
300. if you use “” double quote your don’t need to use \0 null vlaue;
301. for check actual size use: sizeof(name);
302. don’t need array init;
303. **print super power**
304. printf(“%s”,name);
305. dont’t need for loop;
306. if you not set size then automatically set ‘\0’ null of the end; if you set size.you manually add the ‘\0’ null;
307. **input super power**
308. scanf(“%s”,name);
309. don’t need: &,loop
310. but problem is: space is her end line;
311. **input super power with space**
312. fgets(name,size,stdin);
313. fgets take input alson enter key.it’s rember ok.
314. **string length**
315. ‘\0’=null;
316. using for loop:condition: namep[i] != ‘\0’;
317. using string.h: strlen();
318. **Rules**
319. scanf() when get a space he is imagine int’s new line or enter.
320. **Module-10.5: Practice Day**
321. [problem links](https://docs.google.com/document/d/1geZ2RYI0CXKm3O0Y5O9643MR8tlf82G_ozJexc06RrA/edit)
322. **Module-11: String Operation and Freequency Array**
323. **introduction**
324. string copy
325. compare two string
326. concatenation two string
327. counting array
328. **string copy**
329. using logic
330. using strcpy()
331. **string laxicographical comparison**
332. infinity loop declare
333. while(1),for(int I =0; ;i++)
334. condition is uses ASCII value compare
335. condition is uses nul \0 value
336. use strcmp() built in function
337. **string concatenation**
338. **counting or frequency array -1 using number**
339. count how many same value store in array
340. **counting or freequency array -2 using alphabet**
341. it’s amazing
342. just -97
343. **Module-13: Nested loop recape**
344. **Introduction**
345. nested loop
346. different nested loop pattern
347. one problem solve using nested loop
348. nested loop sorting algorithm
349. **nested loop pattern 1**
350. print \* given total time;
351. **nested loop pattern 2**
352. print tringle
353. **nested loop pattern 3**
354. print print tringle
355. **nested loop pattern 4**
356. print diamond
357. **problem solving**
358. sum of two values
359. **sorting alogrithom**
360. ascending or descending sorting an array
361. **Module-14: Function and Pointer**
362. **Introduction**
363. about function
364. different function
365. built in function
366. basic pointer
367. **useful built in function**
368. ceil()
369. floor()
370. round()
371. sqrt()
372. pow()
373. abs()
374. **scope**
375. %p,&a;
376. **pointer**
377. **Module-15: Function With Pointer**
378. **introduction**
379. about pointer
380. about call by value and call by reference
381. array relation with pointer
382. function working with array and string
383. array how to working call by refernce way in a function
384. **pointers in c**
385. refernce by reference
386. \*name used for pointers
387. **pass by value;**
388. pass a value;
389. **pass by reference**
390. pass address
391. **Module-17: Recurson Function**
392. **introduction**
393. call stack
394. recursion
395. print using recursion
396. recusion drive way of reverse
397. print array using recursion
398. find out length of string using recursion
399. **call stack**
400. call stack data structure
401. LIFO
402. wait
403. data structure like
404. call stack = function call like stack structure
405. **what is recursion**
406. self function calling called recursive function
407. **print from 1 to 5 using recursion**
408. base case
409. **print from 5 to 1 using recursion**
410. recursion is infinity
411. **print array using recursion**
412. just need 3 parameter
413. **length of string using recursion**
414. just awesome
415. **Module-18: 2D Array**
416. **declaration**
417. daty\_type array\_name[rows][columns]
418. **introduction**
419. 2D array
420. 2D array input and output
421. matrix and some matrix type
422. profes types using 2D array
423. **what is 2D array?**
424. 2D array like matrix
425. row\*column = [count\_array][size]
426. **2D array input and output**
427. row looping then column looping
428. **types of matrix**
429. row matirx
430. column matrix
431. zero matrix
432. diagonal matrix
433. scaler matrix
434. unit matrix
435. **Module-18.5: Practice Day**
436. absolute number print
437. stdlib.h
438. abs();
439. **Module-19: Recursion Recape**
440. **introduction**
441. mirror or inverted 2D array value;