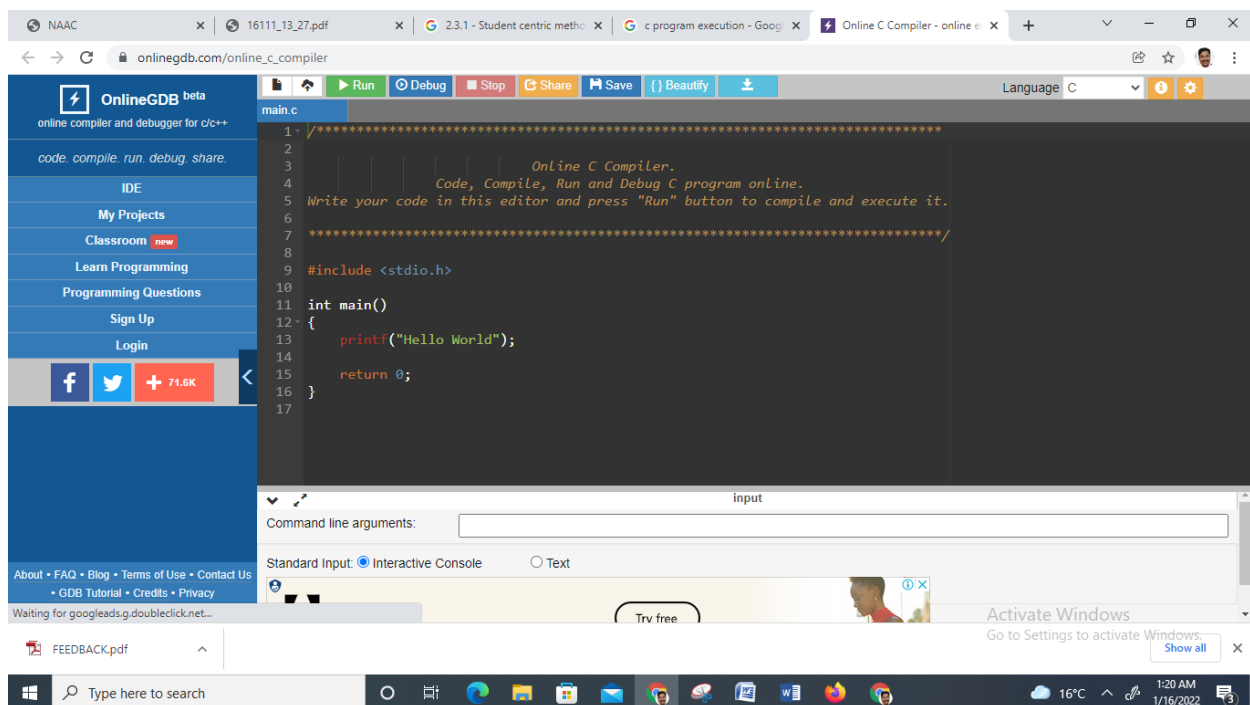


## Problem solving methodologies adopted are

1. Giving regular assignments
2. Quizzes at the end of instruction of each unit.
3. Objective questions are given.
4. Online execution of program by using assemblers
5. Virtual Instrumentation

All academic activities are elevating the students' knowledge, skills and build confidence in them.

## Assemblers are used to execute program



The screenshot displays the OnlineGDB web interface. The browser tabs include 'NAAC', '16111\_13\_27.pdf', '2.3.1 - Student centric metho...', 'c program execution - Goog...', and 'Online C Compiler - online...'. The URL is 'onlinegdb.com/online\_c\_compiler'. The interface features a navigation menu on the left with options like 'IDE', 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Sign Up', and 'Login'. The main editor area shows a C program in 'main.c' with the following code:

```
1- /*****  
2-  
3- Online C Compiler.  
4- Code, Compile, Run and Debug C program online.  
5- Write your code in this editor and press "Run" button to compile and execute it.  
6- *****/  
7-  
8-  
9- #include <stdio.h>  
10-  
11- int main()  
12- {  
13-     printf("Hello World");  
14-  
15-     return 0;  
16- }  
17-
```

Below the code editor, there is an 'input' field for command line arguments, a 'Standard Input' section with radio buttons for 'Interactive Console' (selected) and 'Text', and a 'Trv free' watermark. The Windows taskbar at the bottom shows the search bar, task view, and various application icons, with the system tray displaying '16°C' and '1:20 AM 1/16/2022'.

# Executing unit for 8051 microcontroller

The screenshot shows a Bing search result for "8051 assembler". The main content is a code editor window titled "Thomas@banknecht:~/src/asm51" containing assembly code for an 8051 microcontroller. The code includes directives like `$nois`, `$include`, and `$list`, followed by memory and register manipulations such as `org`, `mov`, `sp`, `pc`, and `mov`. Comments explain the purpose of various instructions, such as setting a stack pointer and clearing bits. The code ends with a `loop:` label and a `hlt` instruction.

```
$nois
#include(c:\reg832.pdf) ; comment
$list

data1 equ    020h

org    000h
org    0000111100001111b
mov    a, @a + @ptr
mov    a, @a
mov    a, @R0
mov    sp, #07fh ; stack pointer, hex value
mov    b, #00001111b ; 8 bits: ck
mov    PC, #012h
mov    PC, #0123d
mov    p6, #000h

; invalid values are not colored
mov    a, #12h ; # must be followed by 0
mov    a, #20d ; # must be followed by 0
org    10000111100001111b ; a bit to much
mov    a, #00001111b ; a bit to much

loop:
hlt
```

Below the code editor, there are navigation buttons for "Visual Search", "Save", "Share", and "More". To the right of the code editor, there is a search bar and a list of related content, including a link to "Thomas' writings, builds, toys | Vim 8051 assembler syntax" and a "Related content" section with several thumbnail images of code editors.