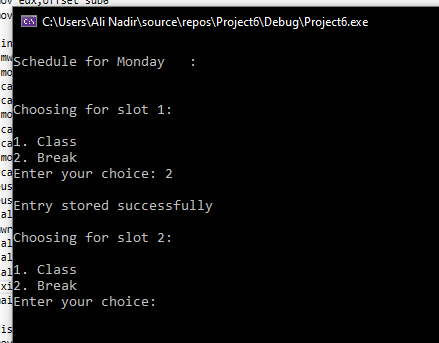
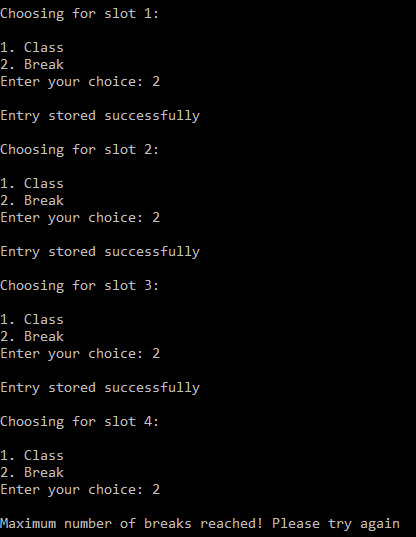
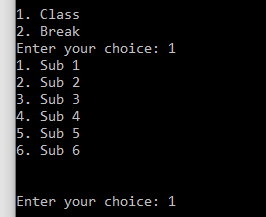
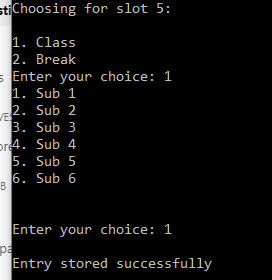
**TESTING TIME AND TIMETABLE FUNCTIONALITY:**

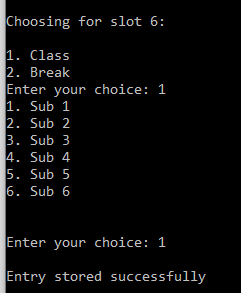
* Displaying the generated timetable after user input
  + Input is taken for every day of the week
  + For every day, there are 8 slots as shown below, so input taken 8 times per day
  + Choice of either break or class is given
  + 
  + If class is selected, user enters choice of subject to be added to slot
  + If break selected, index manipulated in such a way that table displays “FREE” on console
  + For both breaks and a certain subject, a maximum of three breaks and maximum of three slots of the “SAME SUBJECT” are allowed
  + **Same BREAK:**
  + 

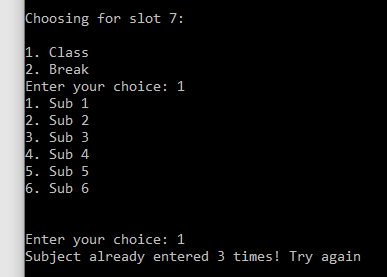
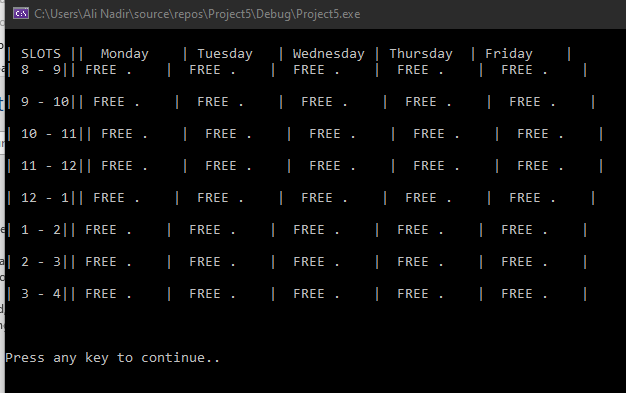
**1**

**2**

**3**

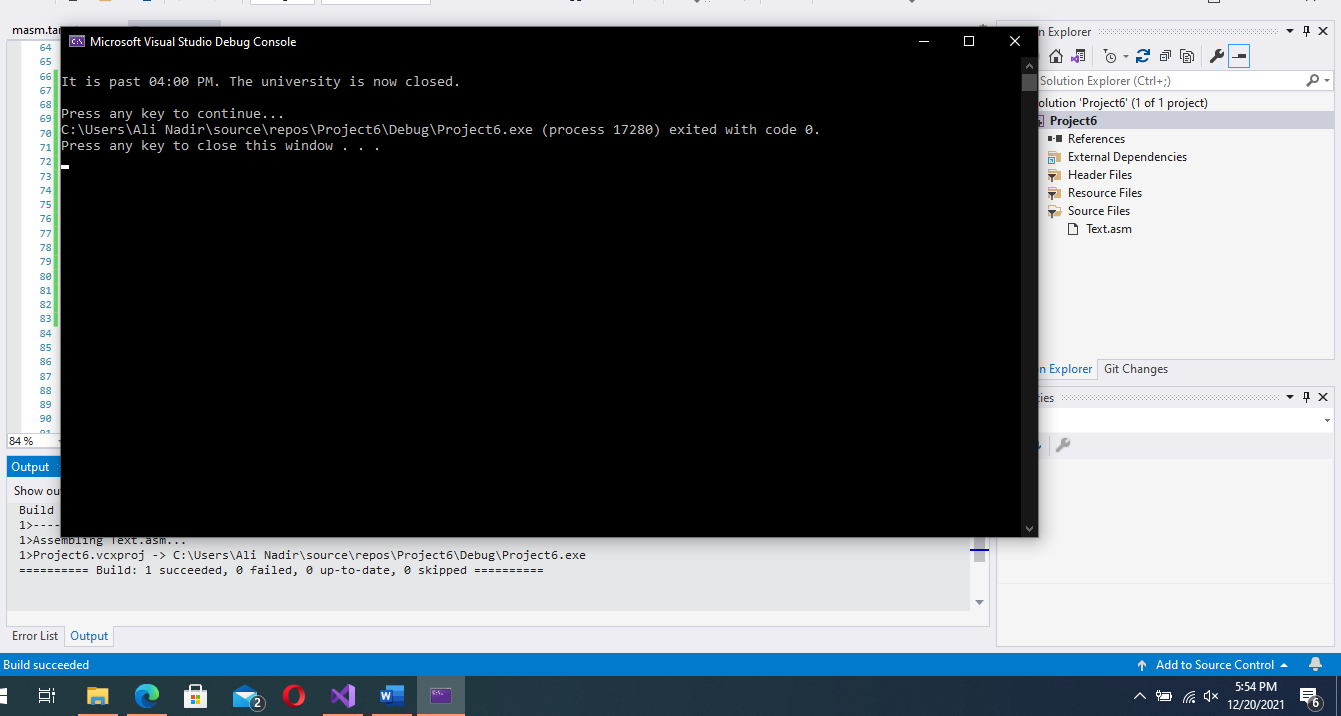
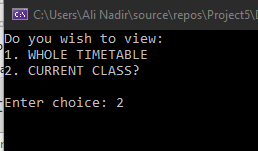
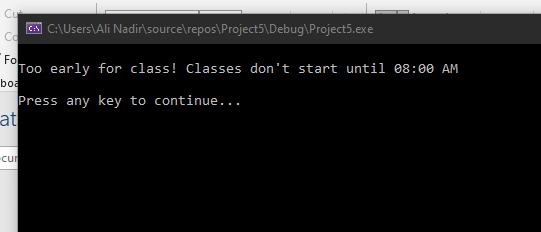
* + 
  + 



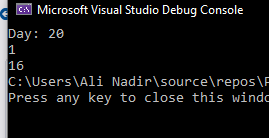
* + 
  + Once timetable is set for all days, table is then displayed
  + For display, ViewTable procedure can be called
  + ViewTable procedure offers user 2 options:
    - Display WHOLE time table
    - Display CURRENT class
  + **Output for WHOLE:**
  + 
  + **Default output: (if no entries have been made to timetable)**
  + ****

**Output for CURRENT:**

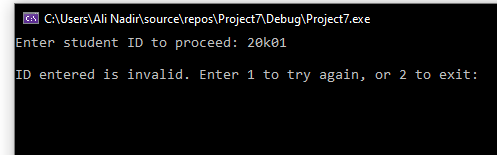
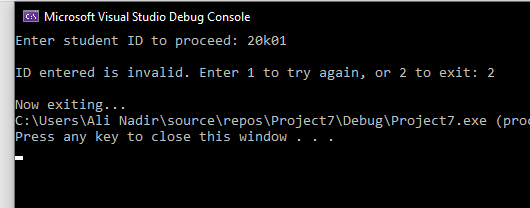
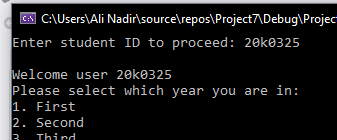
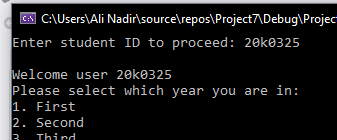
**1.**

* + S
  + S
  + S
  + S
  + S
  + S
  + S
  + S
  + S
  + Time at which routine was tested:
  + W
  + W
  + **2.**
  + 
  + 
  + Time tested:
  + 

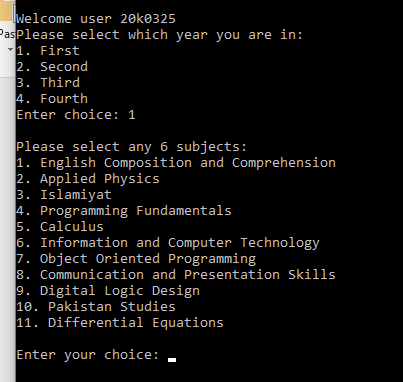
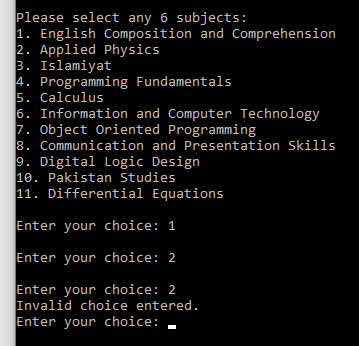
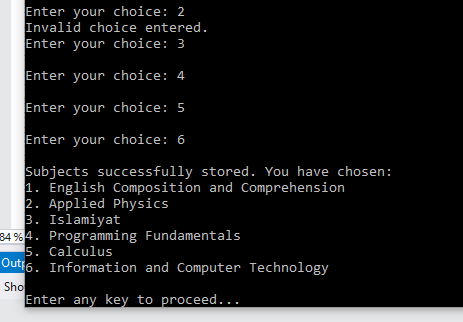
* C**hecking if LPSYSTEMTIME structure from kernel32.lib library is working or not:**
  + First line tests the day number returned by member wDay of structure variable LOCALT
  + Second line shows the day of the week returned by attribute wDayofWeek (1-7) (Tested on Monday so displays 1)
  + Third line shows value of structure attribute wHour (Tested between 4PM-5PM so shows 16 since 16 represents 4th hour after noon (12+4=16)



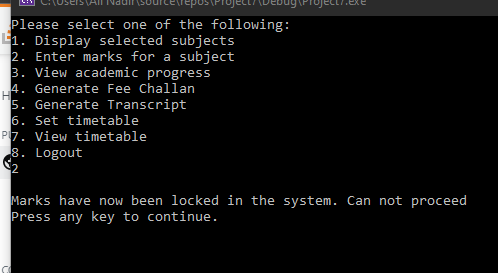
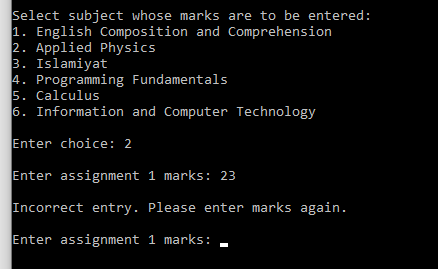
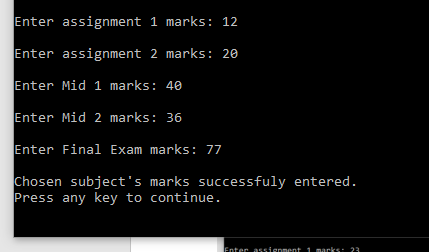
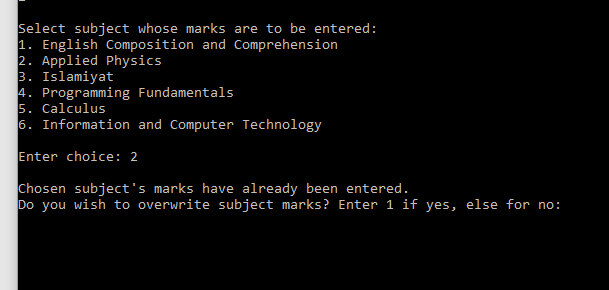
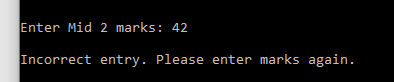
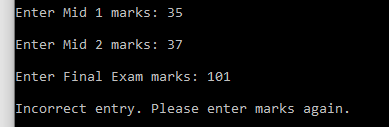
**TESTING STUDENT ID VALIDATION:**

* **Invalid input:**
* 
* **Output:**
* 
* **Valid input:**
* 
* **Ouput:**
* 

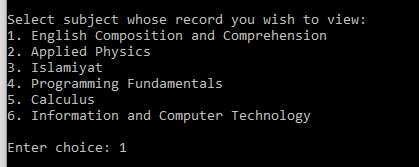
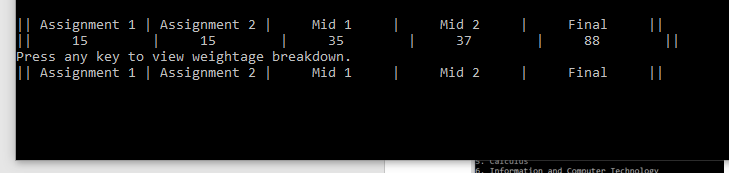
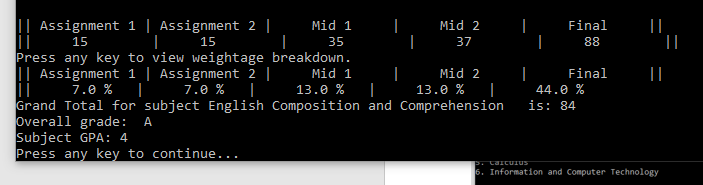
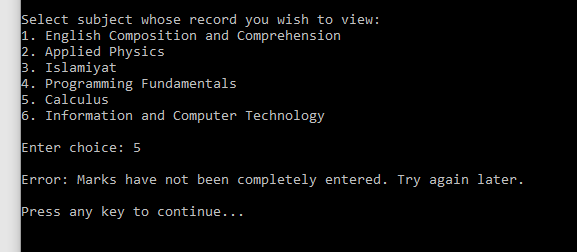
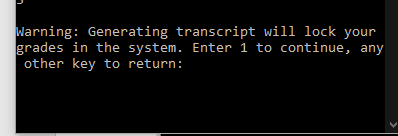
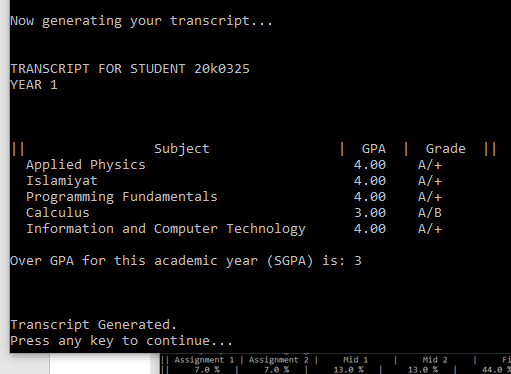
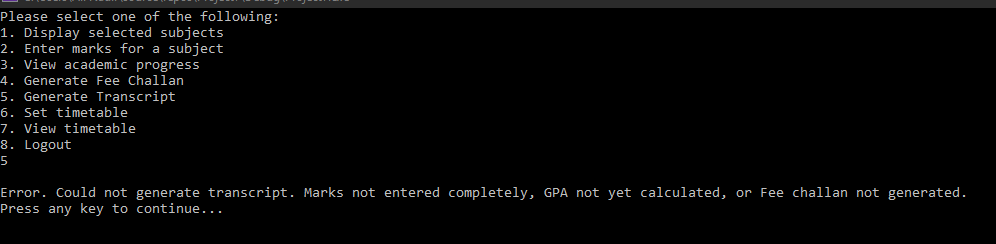
**SUBJECT ENTRY:**

* Entering 1 as year input
* 
* Selecting **6 different** subjects:
* For the same input **twice:**
* s
* S
* S
* S
* S
* S
* S
* S
* Prompts to enter subject again:
* 
* Selected subjects displayed, now pause console output with **readchar**

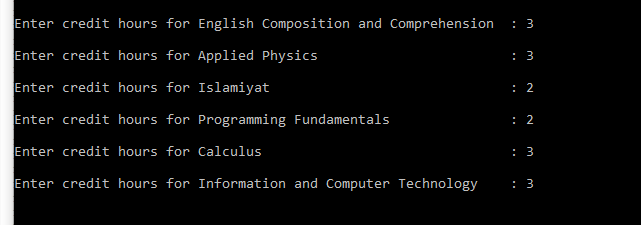
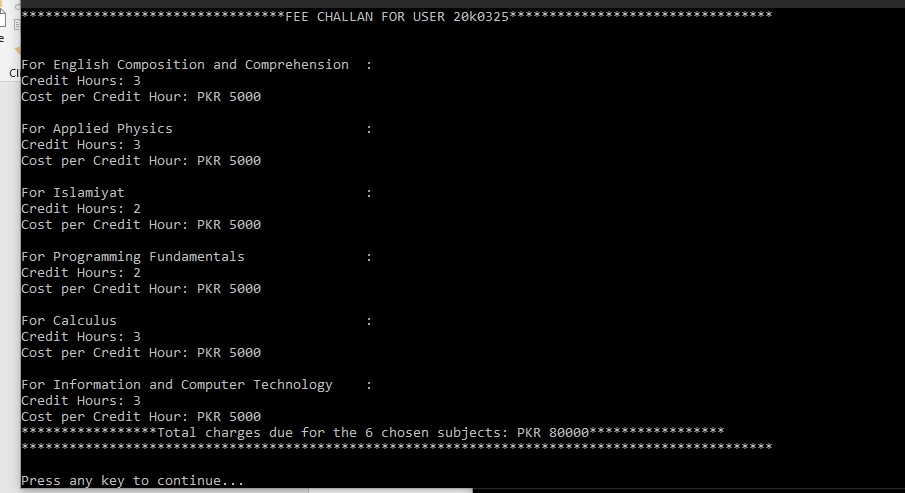
**MARKS ENTRY:**

* If transcript not yet generated, allows marks entry
* Else:
* ****
* Selecting subject no. 2 for entry:
* Entering invalid marks (Assignments are marked out of 20)
* ****
* Prompted for correct entry
* ****
* Redirect to menu after marks have been successfully entered
* If user chooses to reenter marks for an already recorded subject:
  + Ask for confirmation to overwrite marks
  + ****
  + Invalid Mid 1/2 marks (mustn’t be greater than 40)
  + 
  + Invalid final exam marks (mustn’t be greater than 100)

**VIEWING RECORDS:**

* Viewing record for recorded subject: (VALID)
* 
* 
* 
* Invalid choice (subject not recorded yet)
* 
* **GENERATING TRANSCRIPT:**
* Valid output:
  + Ask user for confirmation to lock in grades and proceed with generation
  + ****
  + ****
* Output upon invalidity:
  + ****
  + Requires:
    - Marks to be entered completely
    - GPA to be generated for all subjects (by viewing records at least once)
    - Fee Challan to be generated

**GENERATING FEE CHALLAN:**

* Prompting user to enter credit hours for each subject:
  + ****
* Generated fee challan:
  + ****
* **Total displayed is PKR 80000** for the term
* **Cost for each year:**
  + 1st PKR 5000
  + 2nd PKR 7500
  + 3rd PKR 10000