

CS 101: Computer Programming and Utilization

July-Nov 2017

Prof. Umesh Bellur
(cs101@cse.iitb.ac.in)

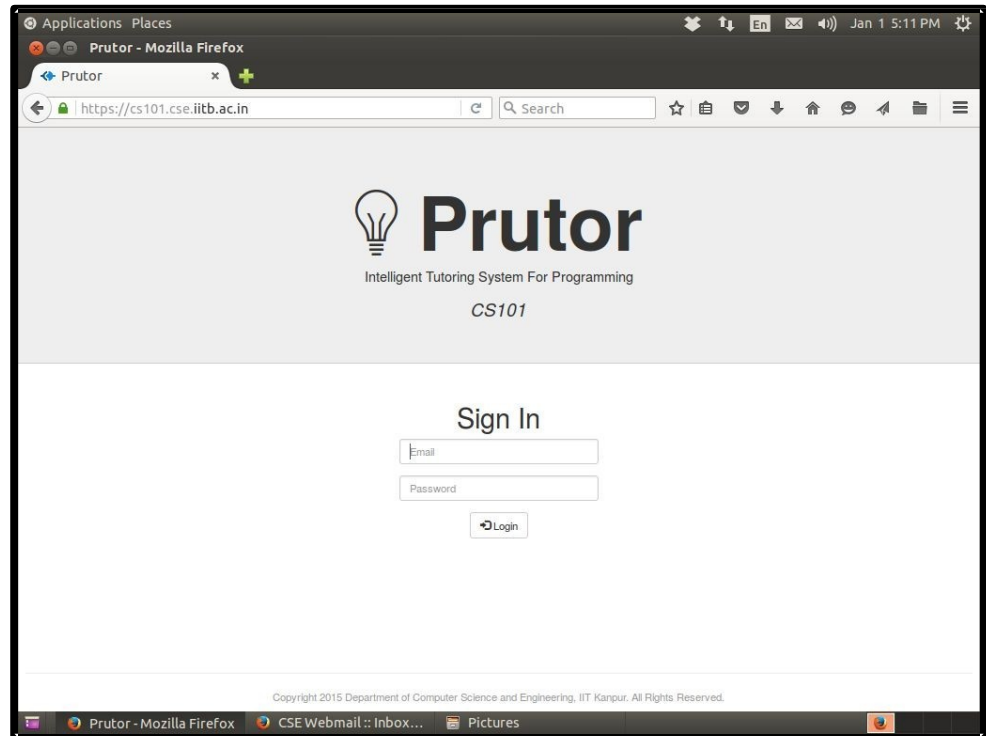
Lecture 1: **Introduction to Prutor**
(Online system for program preparation and submission)

About These Slides

- These slides describe **Prutor** which is a part of an *Intelligent Tutoring System (ITS)* developed by Amey Karkare at IIT Kanpur
- At IIT Bombay, it is available for cs101 at <https://cs101.cse.iitb.ac.in>
- Slides prepared by Nisha Biju and Anshuman Dhuliya

Login Page

- Login at <https://cs101.cse.iitb.ac.in>
- Accesible through the course page also
<http://www.cse.iitb.ac.in/~cs101/>
- Enter your IITB ldap id and password to login
- If Prutor recognizes you as a registered student of cs101, it will authenticate your credentials



Your Homepage

Your homepage shows you the

- **Events** Arena: Questions (i.e. problems for which you need to write programs)
(may be ungraded questions)
- **Course Statistics** Arena: The status of you submissions
- **Grade Card** Arena
- **CodeBook** Arena: Your submitted programs
- **Practice** Arena
- **Scratchpad** Arena

The screenshot displays the Prutor CS101 homepage. The browser window shows the URL <https://cs101.cse.iitb.ac.in/home>. The page features several key sections:

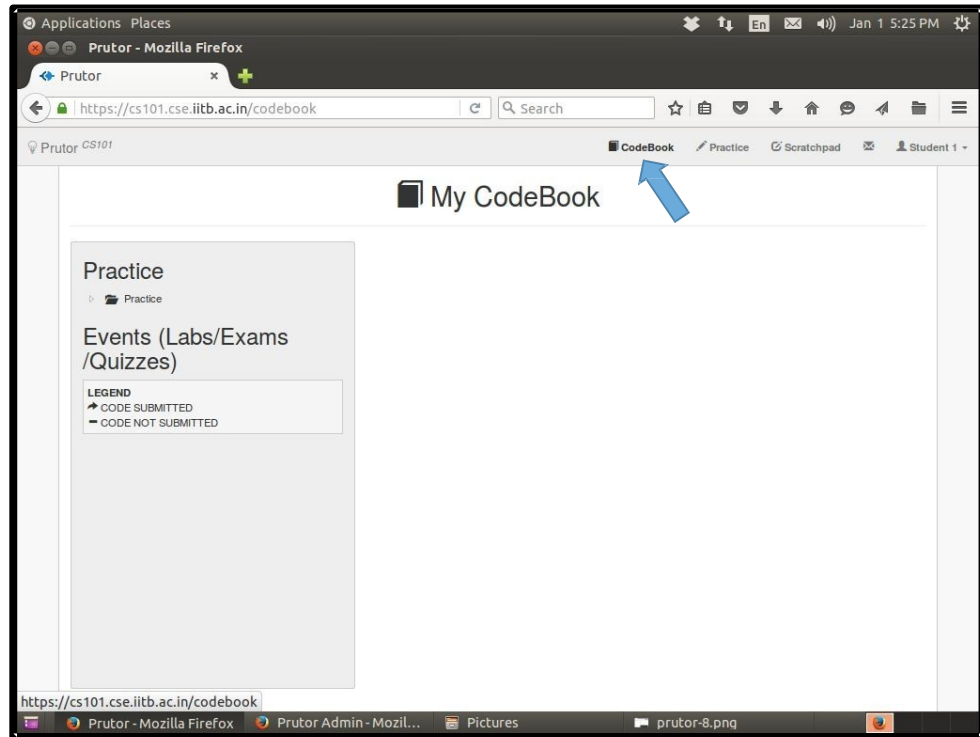
- Ongoing Event:** A section titled "test1" with a deadline of "Ends on Fri Jan 01 2016 at 21:53:00". Below this is a table of questions:

Question ID	Points	Status	Action
Q1	20 Points	submitted	Start Coding
Q1	20 Points	not-submitted	Start Coding
Q2	20 Points	submitted	Start Coding
Q3	20 Points	not-submitted	Start Coding
Q4	20 Points	not-submitted	Start Coding

- Course Statistics:** A section showing "Submitted" (2) and "Not Submitted" (3) counts.
- Course Events:** A section showing "Labs" (1), "Exams" (0), and "Quizzes" (0) counts.
- GRADE CARD:** A button with a document icon and a checkmark, located at the bottom of the main content area.

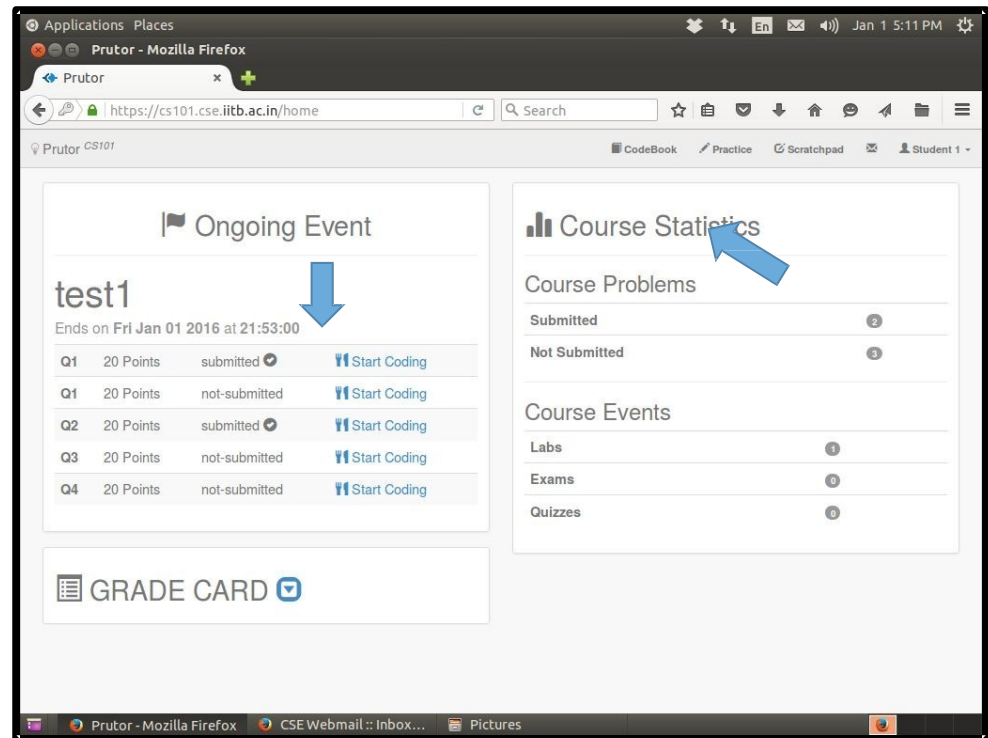
CodeBook Arena

- Here you can see all the problems submitted by you.
- In the picture on the right, the CodeBook is blank as the student has not submitted any question.



Events Arena

- An **Event** is visible only when it is **Ongoing**
 - In the picture on the right, the ongoing event is **test1**, which has five questions with 20 marks each
 - Click on the **Start Coding** link to start solving the questions
 - The events in graded labs will contribute to your grades, and the events in regular labs will be for practice only
- (We will have six graded labs)



The screenshot shows the Prutor CS101 interface in a Mozilla Firefox browser. The page is titled "Prutor CS101" and has a navigation bar with "CodeBook", "Practice", "Scratchpad", and "Student 1". The main content area is divided into two columns. The left column features an "Ongoing Event" section with a flag icon and the title "test1". Below the title, it says "Ends on Fri Jan 01 2016 at 21:53:00". A table lists four questions (Q1, Q1, Q2, Q3, Q4) with 20 points each. The status of each question is either "submitted" or "not-submitted", and each has a "Start Coding" link. A blue arrow points to the "Start Coding" link for the first "not-submitted" question. The right column features a "Course Statistics" section with a bar chart icon. It shows "Course Problems" with "Submitted" (2) and "Not Submitted" (3). Below that, "Course Events" are listed: "Labs" (1), "Exams" (0), and "Quizzes" (0). A blue arrow points to the "Course Statistics" title. At the bottom of the main content area, there is a "GRADE CARD" button with a checkmark icon. The browser's address bar shows "https://cs101.cse.iitb.ac.in/home". The system tray at the bottom shows the time as "Jan 1 5:11 PM" and includes icons for "Prutor - Mozilla Firefox", "CSE Webmail: Inbox...", and "Pictures".

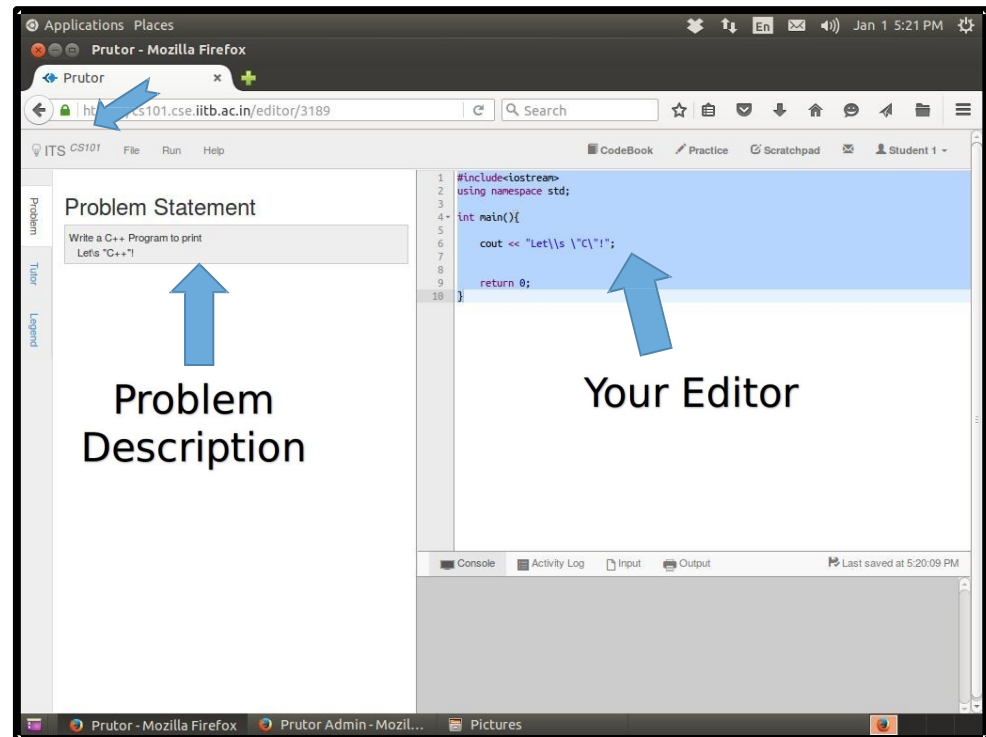
Plain C++ and Simplecpp

- The events arena shows the list of problems
- Some of these are **plain C++** problems and some are **simplecpp** problems
- All the problems are **edited**, **compiled**, and **executed** in the **same manner**
- The **output** for **plain C++** programs appears **directly**
- Viewing the **output** of **simplecpp** needs an **additional step** (Explained in a later slide)

Code Editor

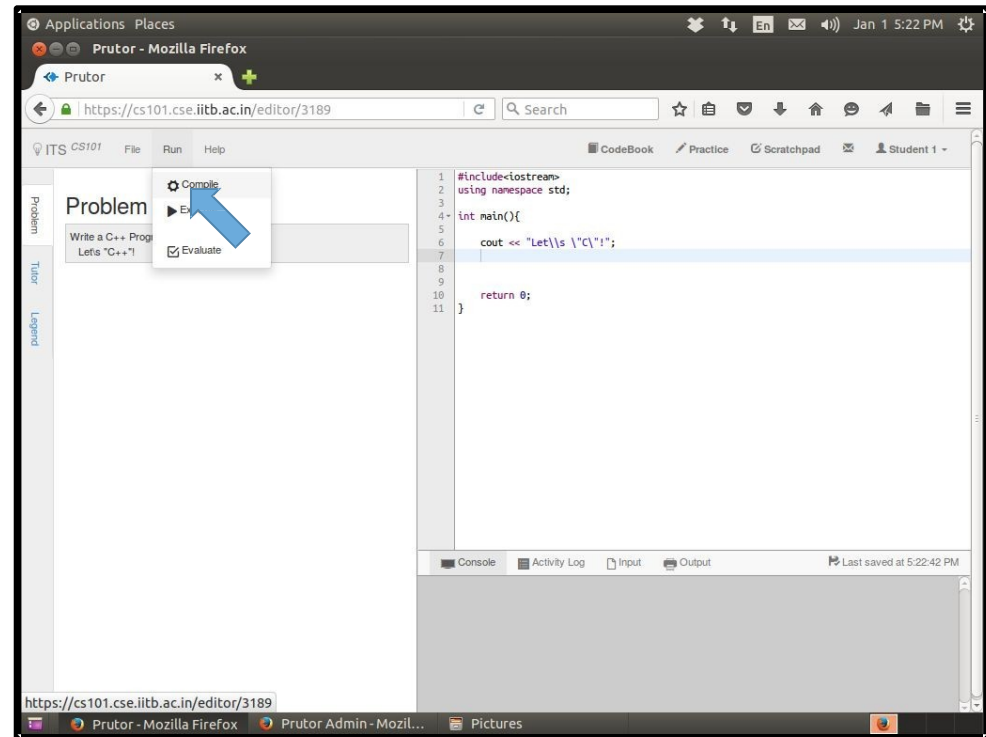
- You reach here by clicking the **Start Coding** link of a question
- The **Code Editor** saves your code every 5 seconds
- You can also save your code by pressing **Ctrl-s**
- Or you can save using the **Save** option of the **File** tab

We can play back your key strokes and retrace your steps to figure out whether you wrote the code yourself or copied it



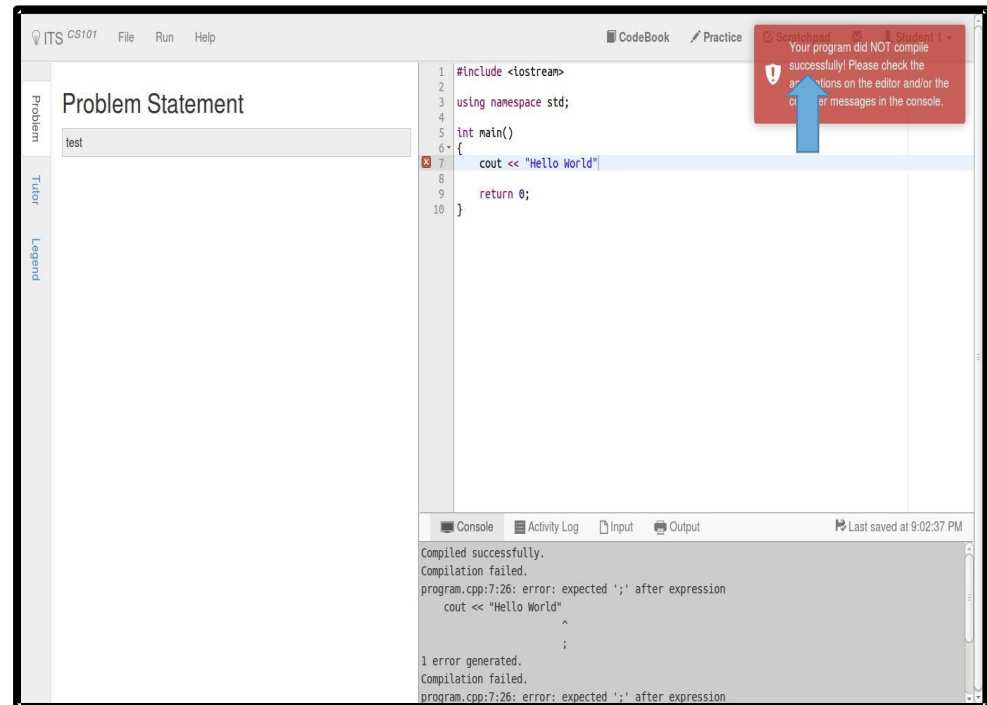
Compiling Code

- After writing the code, choose the **Compile** option in the **Run** tab



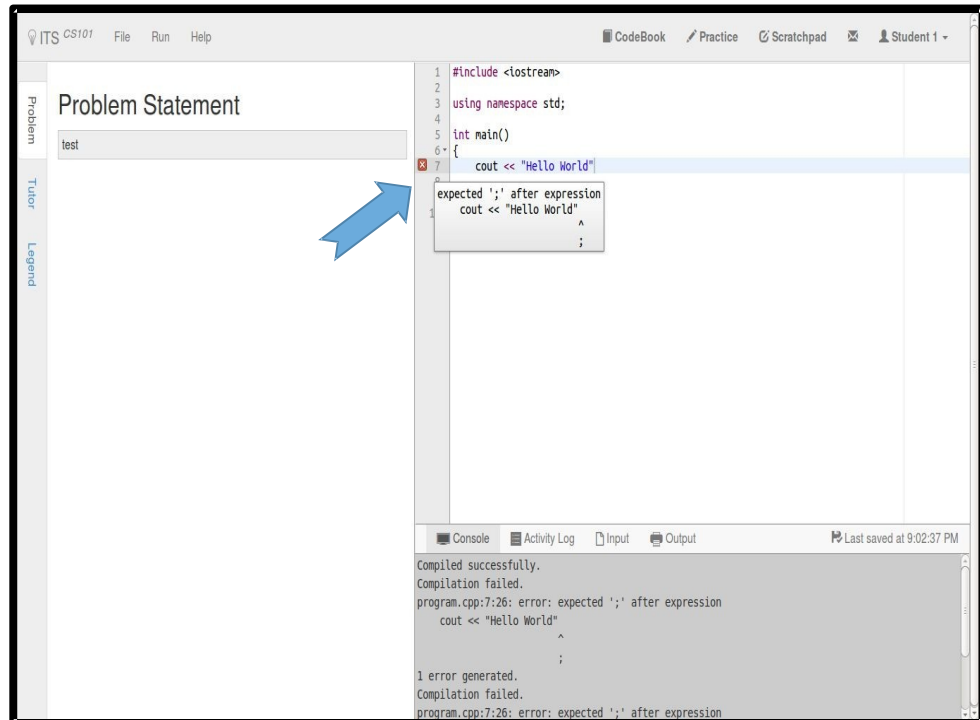
Compiler Messages

- You can view the compilation results below the **Code Editor** in the gray area, which is your virtual console
- Also a **floating notification** is shown on top-right corner for few seconds
- In the picture on the right, the compilation has failed as can be easily inferred from the message in **red blob of the floating notification**



Error Messages

- The compiler also issues messages about errors in specific lines
- We can see the error in the box if we take the mouse pointer towards the left side of the editor
- Read the compiler message at the bottom carefully and correct your error



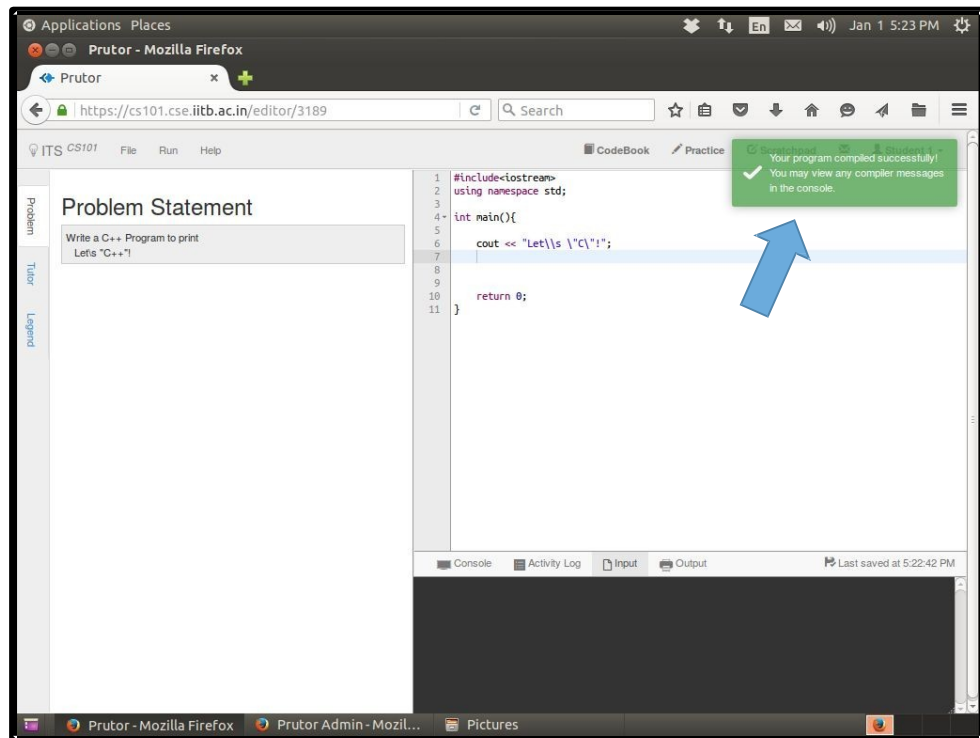
```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     cout << "Hello World"
8     ;
9 }
```

expected \';' after expression
cout << "Hello World" ^
;

Compiled successfully.
Compilation failed.
program.cpp:7:26: error: expected \';' after expression
cout << "Hello World"
^
;
1 error generated.
Compilation failed.
program.cpp:7:26: error: expected \';' after expression

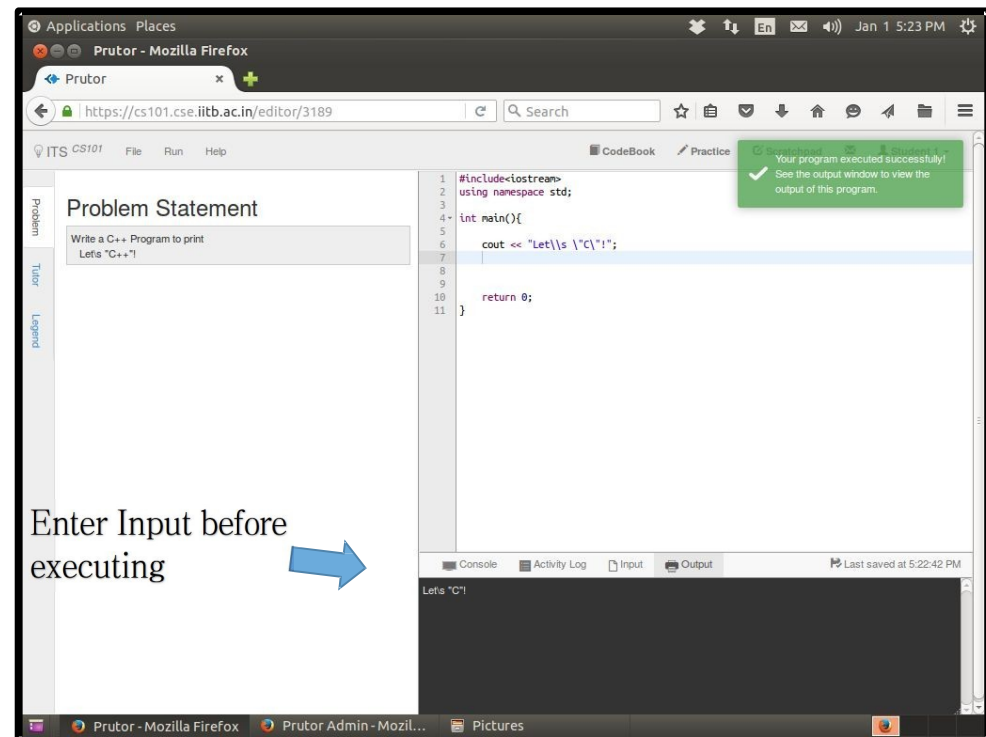
Successful Compilation

- The floating notification on top-right corner will notify you when your compilation is successful
- The color of the floating notification would be green



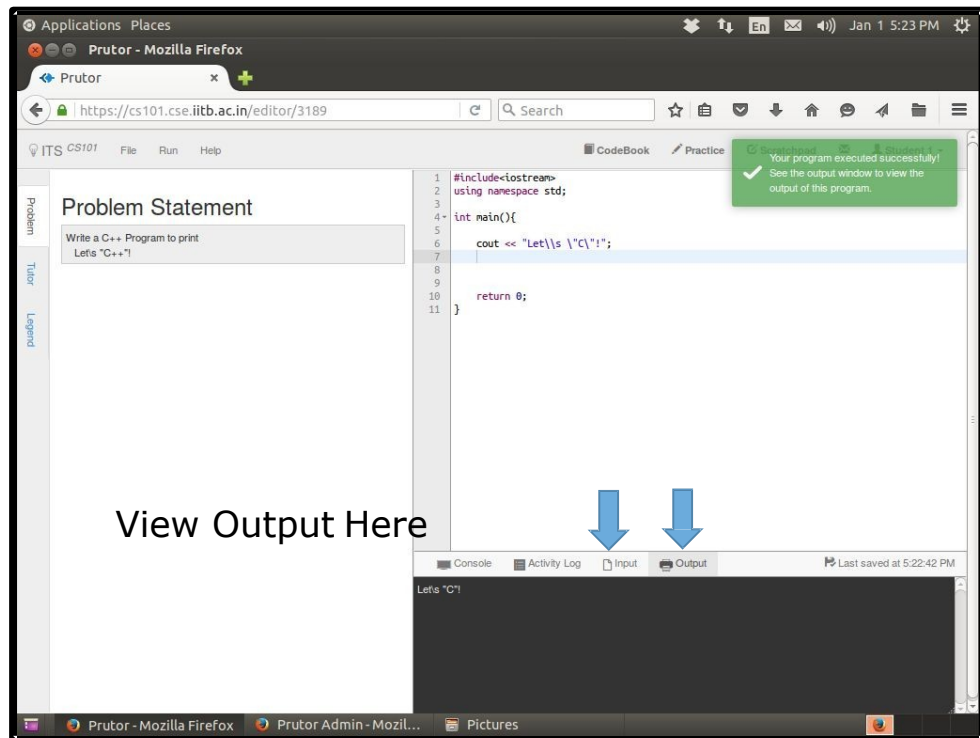
Executing Your Program

- If your program requires input data, then you need to provide it before executing the program.
- For providing the input, click on the **Input** tab below the **Code Editor** and enter the input as you expect it to be
- Click on **Execute** option in the **Run** tab to execute the program
- Note: Before executing make sure that your program has compiled successfully



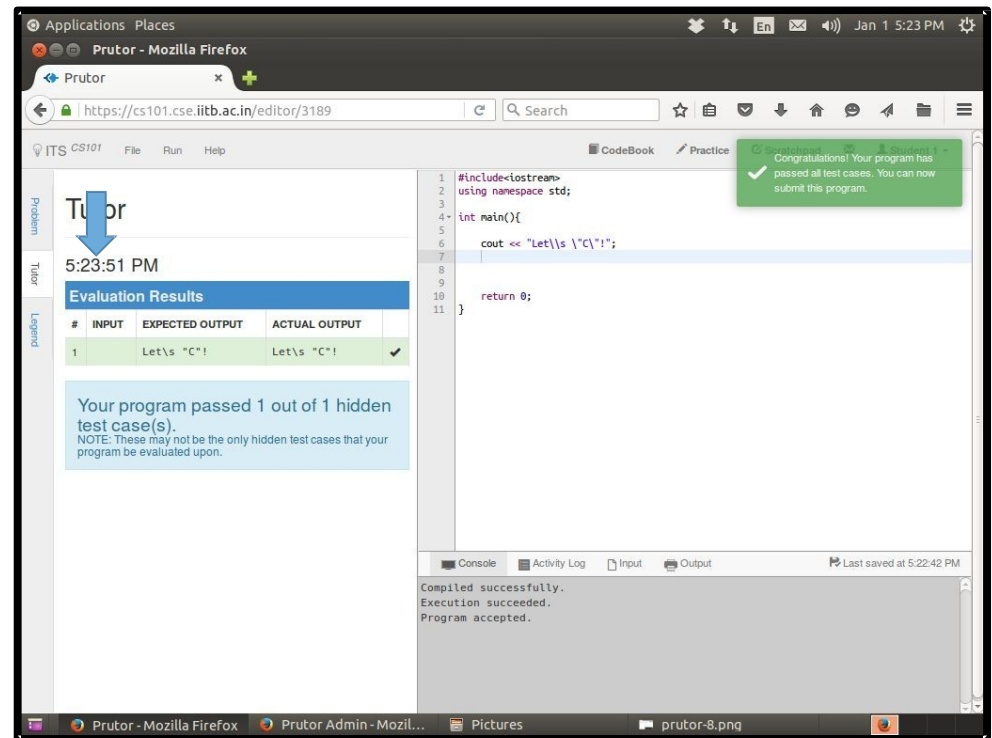
Viewing the Output

- After successful execution, a **green floating notification** will appear on the **top-right corner** and the output will be shown at the **bottom**



Evaluating Your Program

- **Evaluation** refers to running the program on the given test cases
- You can see some test case but there may be some hidden test cases
- Click on the **Evaluate** option in the **Run** tab
- We may evaluate your program on additional test cases created after submission
- Evaluation Results are displayed on the left column heading **Tutor**
- Note: You can submit your code even if it does not pass all the test cases.



Repeated Evaluation

- You can evaluate your program as many times as you like before submission
- This allows you to keep correcting your program
- If your program passes all test cases, a **green floating notification** on the top-right corner will appear with an appropriate message

The screenshot shows the Prutor online IDE interface. The browser address bar displays `https://cs101.cse.iitb.ac.in/editor/3189`. The IDE title is "ITS CS101". The code editor contains the following C++ code:

```
1 #include<iostream>
2 using namespace std;
3
4 int main(){
5
6     cout << "Let\\s \\c"!;
7
8
9
10
11 }
return 0;
```

The console shows the following output:

```
Compiled successfully.
Execution succeeded.
Program accepted.
Program accepted.
```

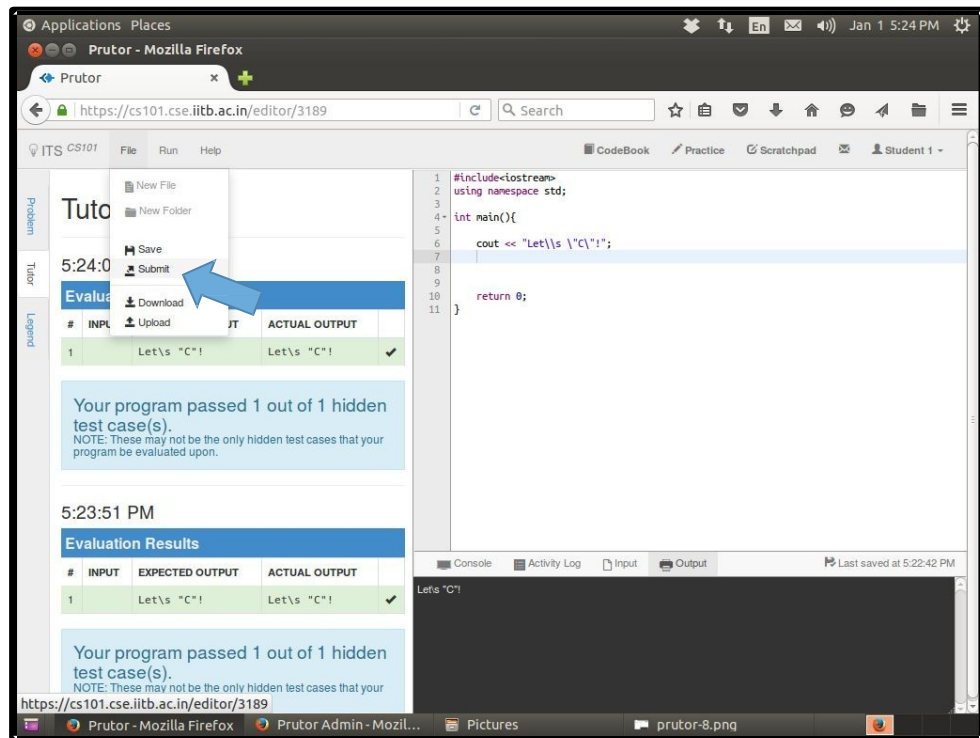
The evaluation results section shows a table with the following data:

#	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	Let\\s \\c"!;	Let\\s \\c"!;	Let\\s \\c"!;

A green floating notification in the top-right corner reads: "Congratulations! Your program has passed all test cases. You can now submit this program." The bottom status bar indicates "Last saved at 5:22:42 PM".

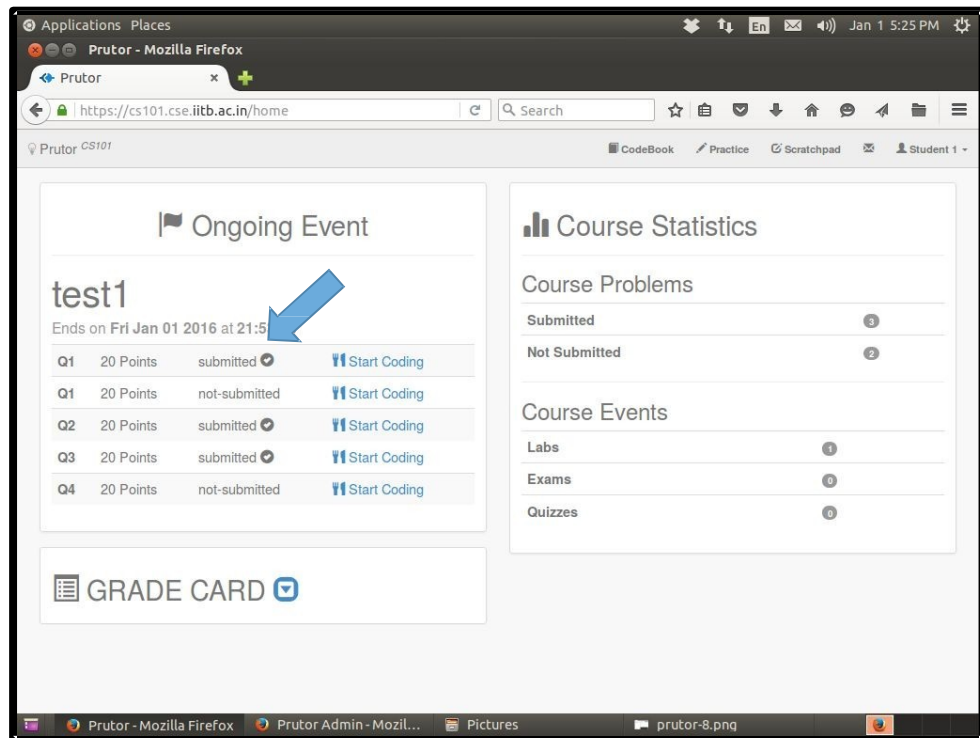
Submit Your Code

- Never forget to submit your code, even if it is incomplete, before the event ends
- However, it is not necessary to submit a code to start attempting another question. But do save it before leaving
- Code that is not submitted can not be evaluated
- To submit, click on **Submit** option in the **File** tab



Homepage After Submission

- After submitting your code, the homepage will mark your question as **submitted**



The screenshot shows the Prutor CS101 homepage in a Mozilla Firefox browser. The page features a navigation bar with 'CodeBook', 'Practice', 'Scratchpad', and 'Student 1'. The main content area is divided into several sections:

- Ongoing Event:** A section titled 'test1' with a sub-header 'Ends on Fri Jan 01 2016 at 21:5...'. Below this is a table of questions:

Question ID	Points	Status	Action
Q1	20 Points	submitted	Start Coding
Q1	20 Points	not-submitted	Start Coding
Q2	20 Points	submitted	Start Coding
Q3	20 Points	submitted	Start Coding
Q4	20 Points	not-submitted	Start Coding

A blue arrow points to the 'submitted' status of the first question (Q1). Below the table is a 'GRADE CARD' button.

- Course Statistics:** A section with a bar chart icon, containing a table for 'Course Problems':

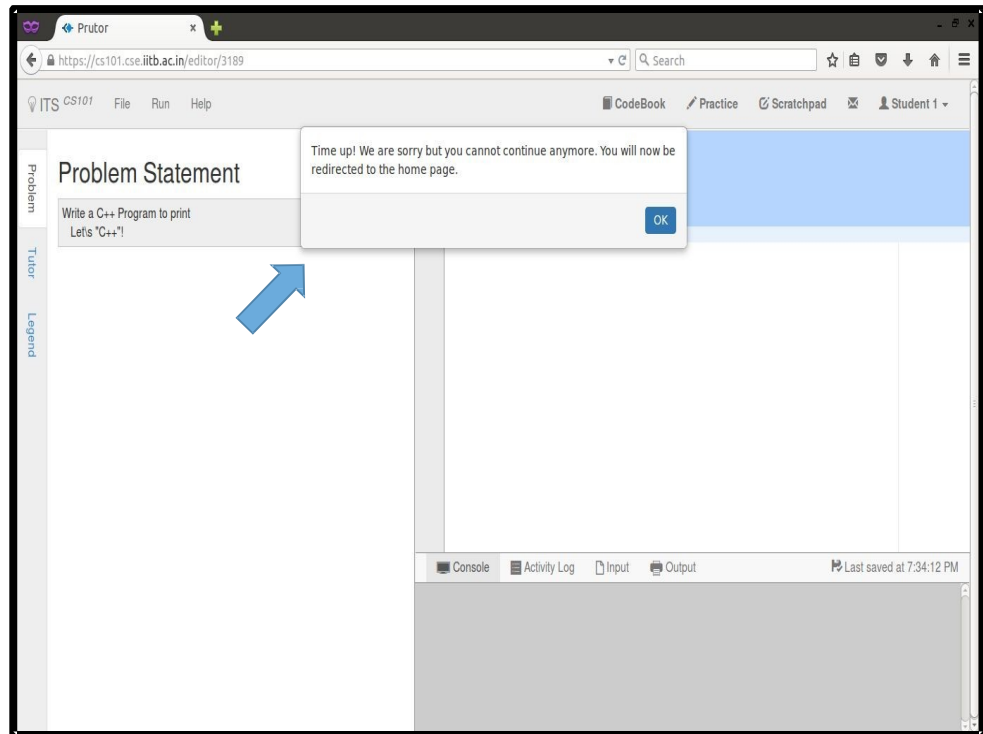
Category	Count
Submitted	3
Not Submitted	2

- Course Events:** A section with a table for 'Course Events':

Category	Count
Labs	1
Exams	0
Quizzes	0

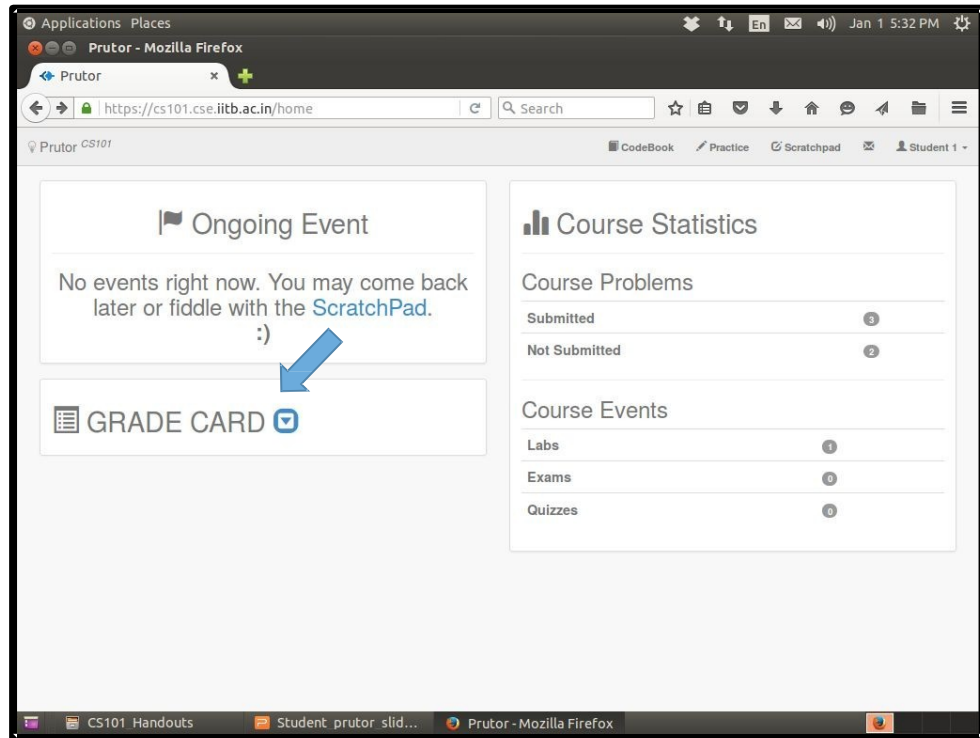
End of Event (1)

- You have limited time to write the code for all questions.
- You will be intimated through a floating notification when you have only 30 minutes left
- When the **time is up**, you will get a notification and you will **not be allowed to edit the code any more**
- You will be redirected to your homepage



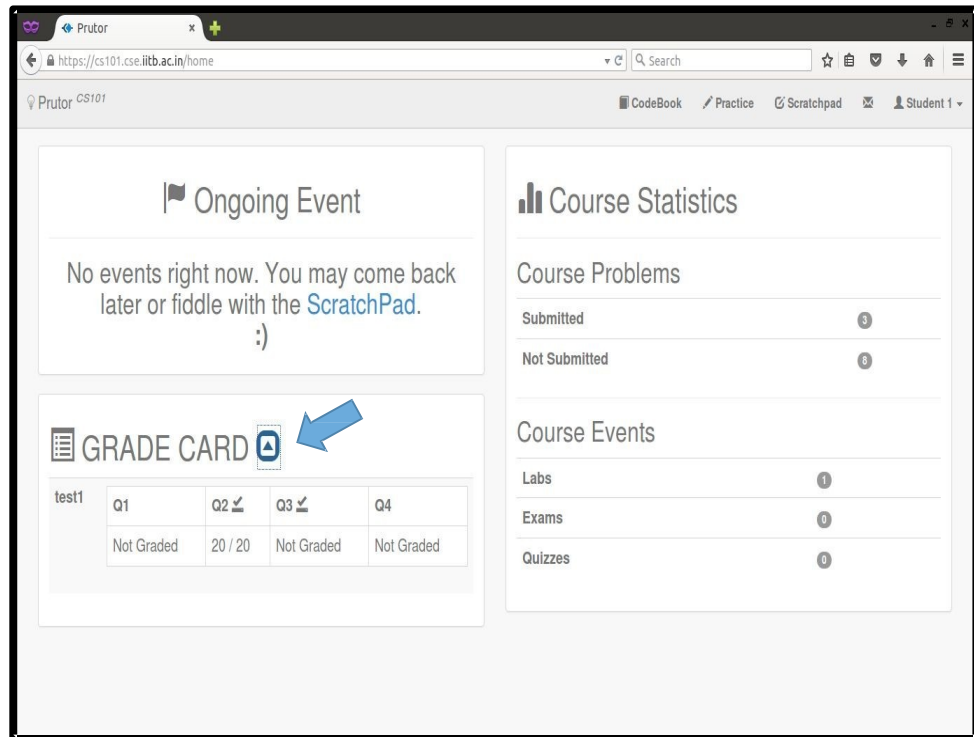
End of Event (2)

- After an event finishes, it stops showing on your homepage (i.e. it is not accessible to you)
- You can view your submitted code after the lab week is over (i.e. all batches have finished the same lab) by visiting your **Codebook**
- Once the grading is done, you can view your marks by clicking on the arrow button next the **Grade Card**



Graded Questions

- When your question is graded the marks are shown in the **GRADE CARD** box
- Only the Q2 is evaluated in this screenshot. The other questions are yet to be evaluated (or the student did not submit the code)



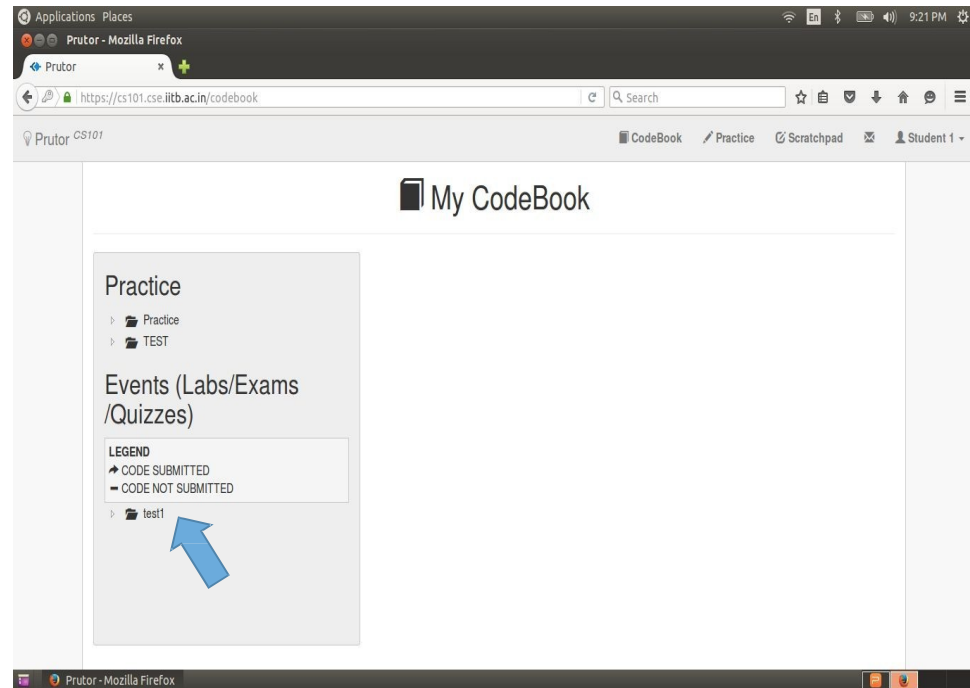
The screenshot shows the Prutor CS101 interface. At the top, there is a navigation bar with 'CodeBook', 'Practice', 'Scratchpad', and 'Student 1'. The main content area is divided into three sections:

- Ongoing Event:** A message stating 'No events right now. You may come back later or fiddle with the ScratchPad.' with a smiley face icon.
- Course Statistics:** A bar chart showing 'Submitted' (3) and 'Not Submitted' (0) for Course Problems, and 'Labs' (1), 'Exams' (0), and 'Quizzes' (0) for Course Events.
- GRADE CARD:** A table showing the results for 'test1'. A blue arrow points to the 'Q2' column, which is highlighted. The 'Q1' column is 'Not Graded', 'Q2' is '20 / 20', 'Q3' is 'Not Graded', and 'Q4' is 'Not Graded'.

test1	Q1	Q2	Q3	Q4
	Not Graded	20 / 20	Not Graded	Not Graded

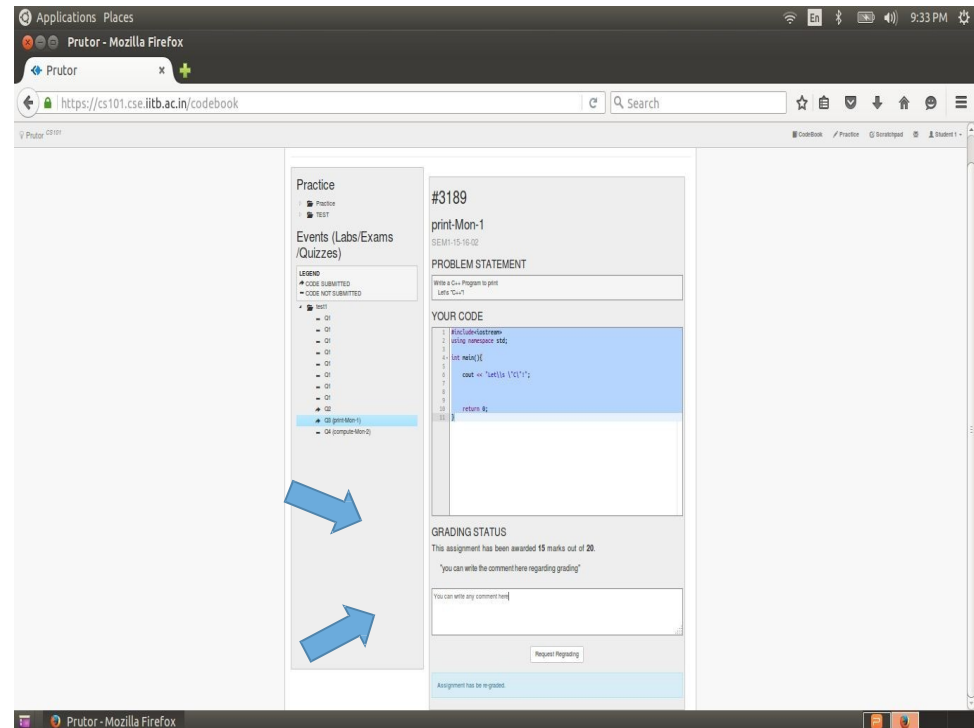
CodeBook (1)

- Here you can see the questions you have attempted in the **test1**
- Click on the question to see your submitted code and the **evaluator's remarks**, if any



CodeBook (2)

- This page gives the details of the problems attempted and grading status, if the problem is part of a **GradedLab**
- You can view the evaluator's remark at the bottom
- In the last text box you can send a **Request** for re- evaluation of your code
- Re-evaluation request can be made only once for a question



The screenshot displays the Prutor CodeBook interface in a Mozilla Firefox browser. The page shows a problem statement for a C++ program, the user's submitted code, and the current grading status. The grading status indicates that the assignment has been awarded 15 marks out of 20. A text box at the bottom allows the user to write a comment regarding grading, and a 'Request Re-grading' button is visible. Two blue arrows point to the 'Request Re-grading' button and the comment text box.

Applications Places
Prutor - Mozilla Firefox
Prutor
https://cs101.cse.iitb.ac.in/codebook

Prutor CS101

Practice
Events (Labs/Exams/Quizzes)
LEGEND
CODE SUBMITTED
CODE NOT SUBMITTED

#3189
print-Mon-1
SEM1-15-16-02
PROBLEM STATEMENT
Write a C++ Program to print
"Sat! 5-4"

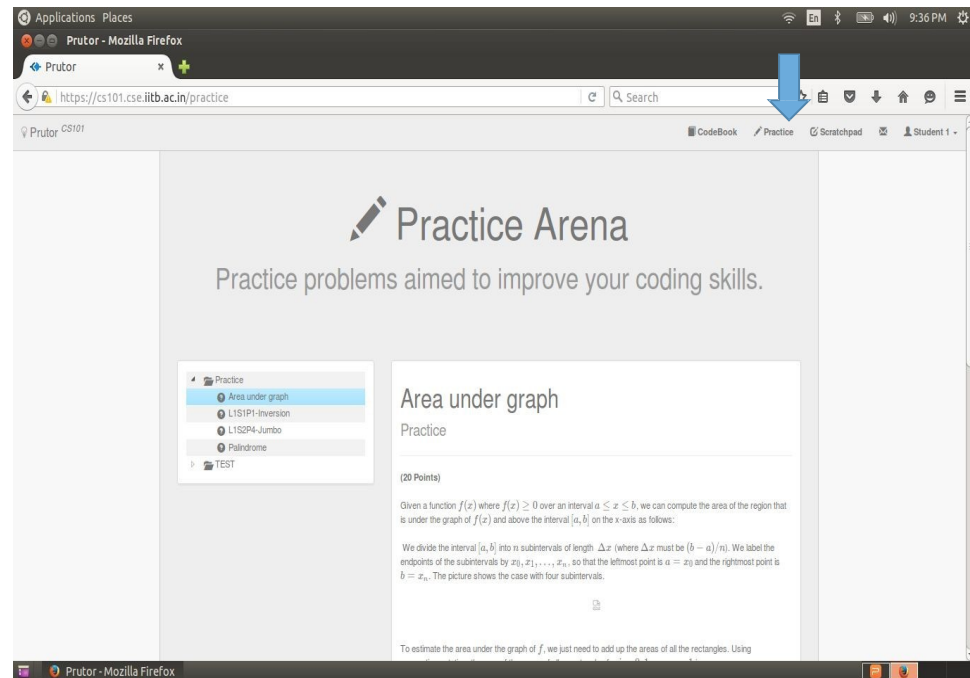
YOUR CODE

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Sat! 5-4";
    return 0;
}
```

GRADING STATUS
This assignment has been awarded 15 marks out of 20.
"You can write the comment here regarding grading"
You can write any comment here
Request Re-grading
Assignment has to be re-graded.

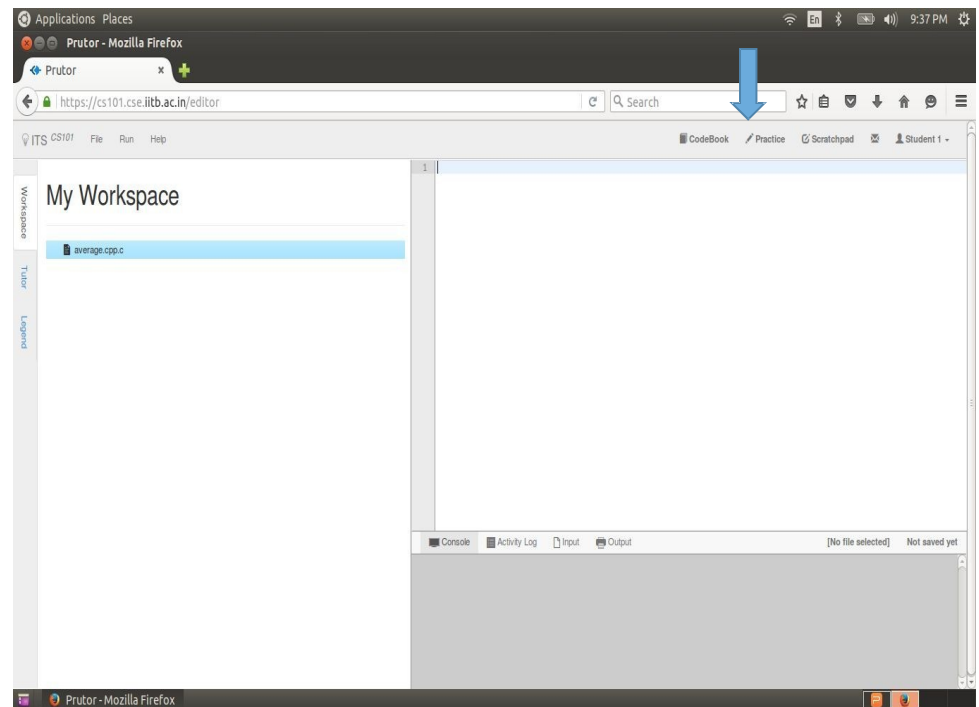
Practice

- You can use the **Practice Arena** to practice programming
- Here you get the same Code Editor where you can **Compile**, and **Execute** your code.
- This section will contain practice problems (along with test cases, if any)
- The code in the Practice Arena is not for submission



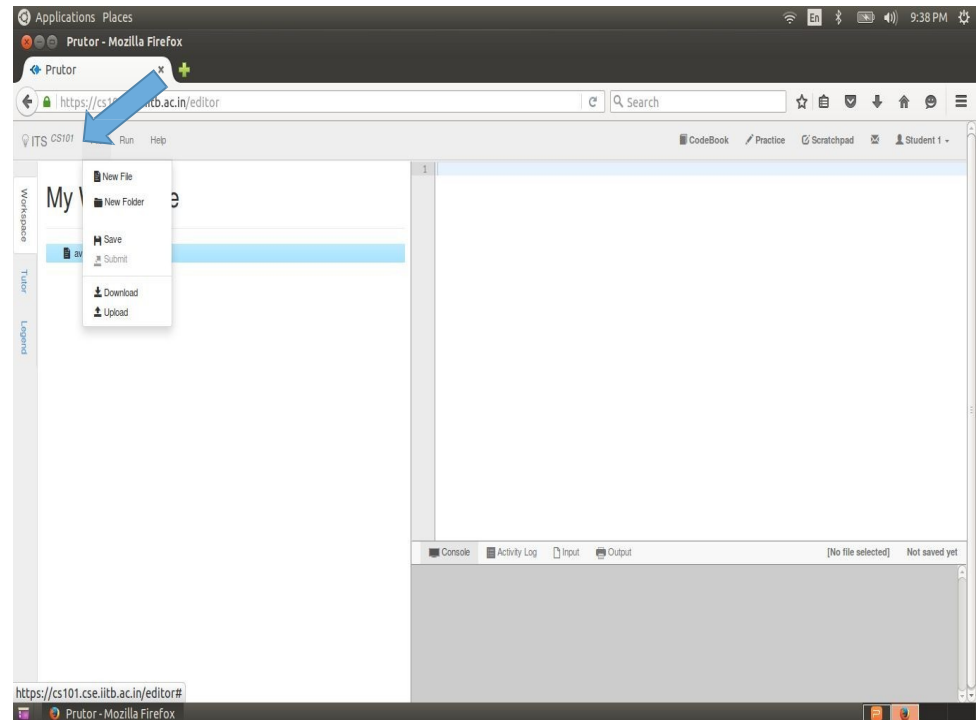
Scratchpad (1)

- You can use the **Scratchpad** to practise programming on your own (no questions are provided)
- You need to create a file before you can start editing
- Here you get the same **Code Editor** where you can **Compile**, and **Execute** your code



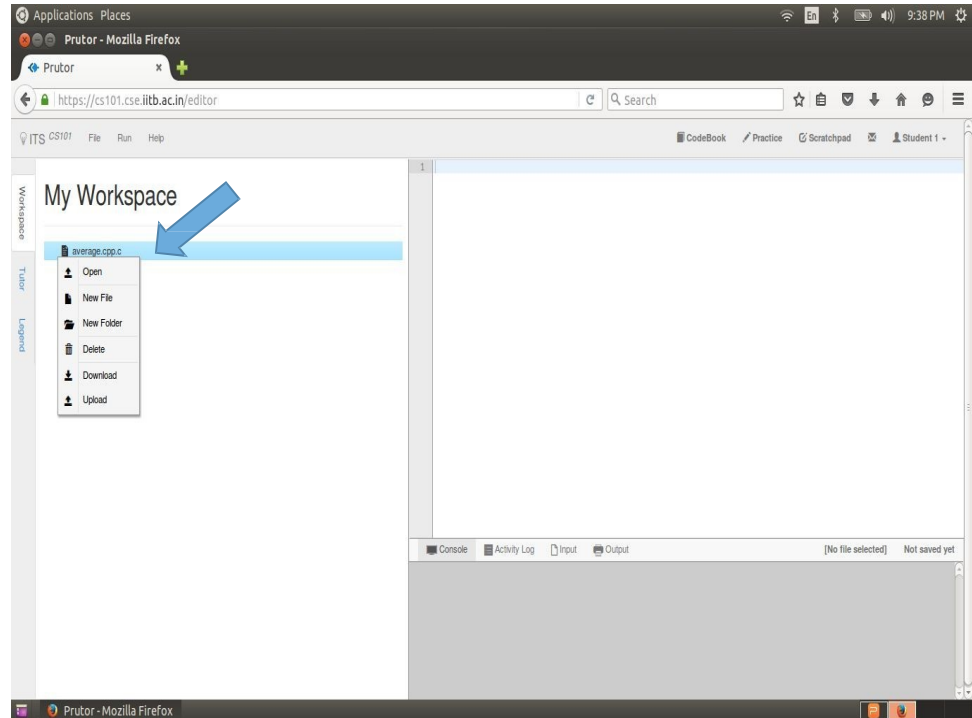
Scratchpad (2)

- You can add files and directories in the **Workspace** using the **File** tab
- If you don't know what files and directories are, don't worry right now. You will learn towards the end of the course. They are not needed right now



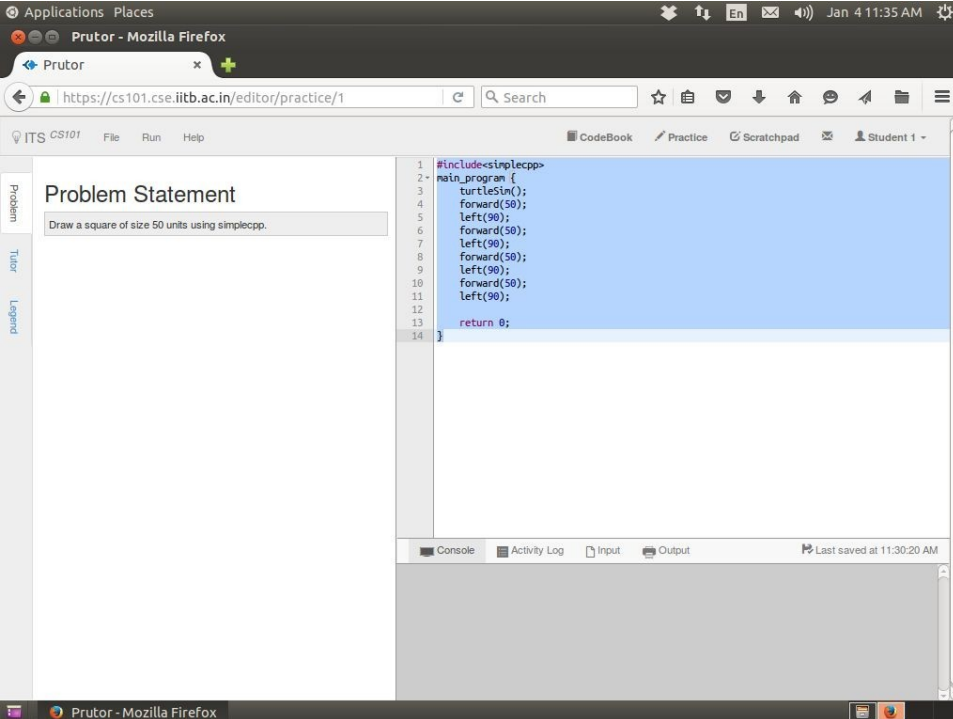
Scratchpad (3)

- You can **add** files and directories to a directory created earlier by using a right click on the names in the **Workspace**



Creating Simplecpp Programs

- You reach here by clicking the **Start Coding** link of a question
- The **Code Editor** saves your code every 5 seconds
- You can also save your code by pressing **Ctrl-s**
- Or you can save using the **Save** option of the **File** tab



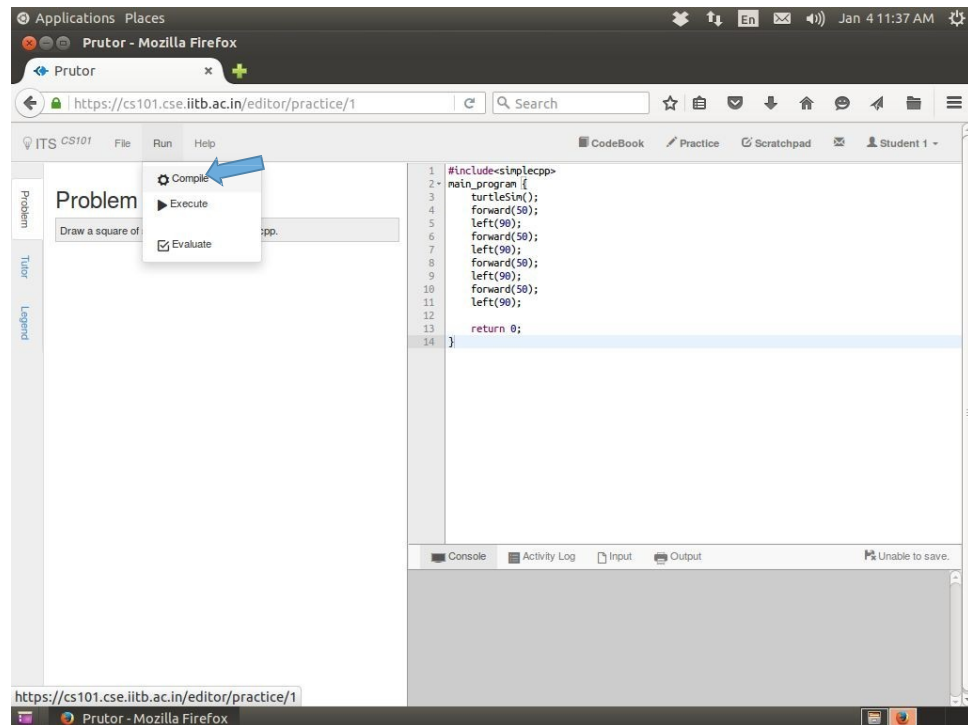
The screenshot shows the Prutor online code editor interface. The browser address bar displays the URL `https://cs101.cse.iitb.ac.in/editor/practice/1`. The page title is "ITS CS101". The interface is divided into several sections:

- Problem Statement:** Located on the left, it contains the text "Draw a square of size 50 units using simplecpp."
- Code Editor:** The main area on the right, containing the following C++ code:

```
1 #include<simplecpp>
2 main_program {
3     turtleSin();
4     forward(50);
5     left(90);
6     forward(50);
7     left(90);
8     forward(50);
9     left(90);
10    forward(50);
11    left(90);
12
13    return 0;
14 }
```
- Console/Output:** At the bottom, it shows "Last saved at 11:30:20 AM".

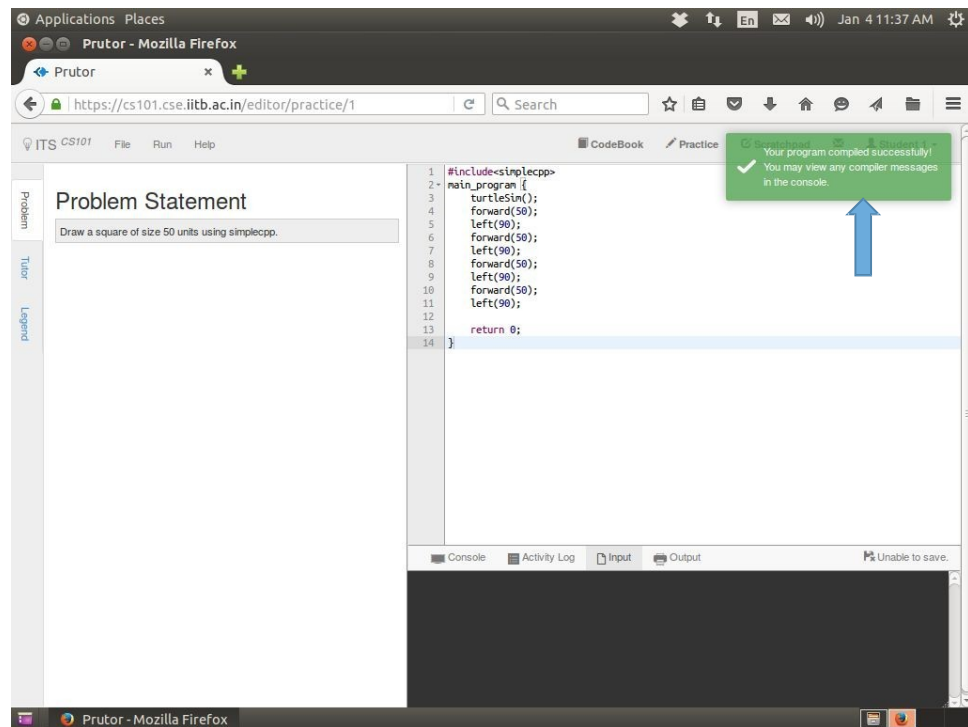
Compiling Simplecpp Programs

After writing the code, choose the **Compile** option in the **Run** tab



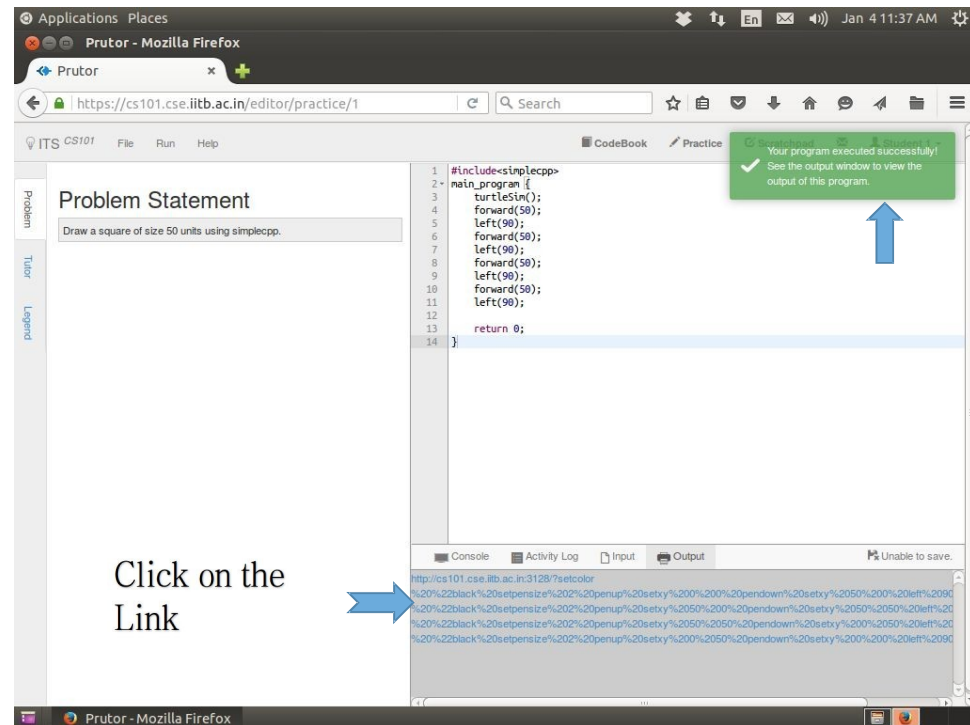
Successful Compilation

- The floating notification on top-right corner will notify you when your compilation is successful
- The color of the floating notification would be green



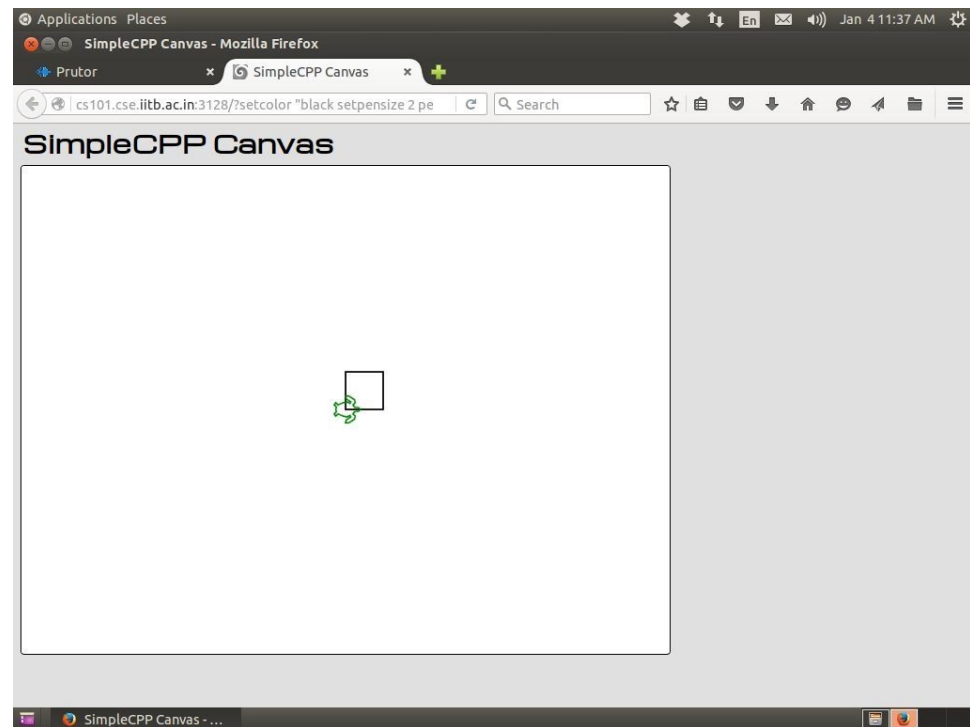
Executing Simplecpp Programs

- Click on **Execute** option in the **Run** tab to execute the program
- After successful execution, a **green floating notification** will appear on the **top-right corner** and a **link** will be shown at the **bottom**
- Click on the **link** in the output window to view the **output**



Viewing the Output of Simplecpp Programs


- You reach here by clicking the link in the output window



Prutor Does Not Support Interaction

- We cannot interleave input and output
- We cannot expect to see the output **No. of sides?** and give the input after it
- The console window is first used as an input window and then the same window is used as an output window
- Do not print any thing extra in the input window or console other than the input needed by the program
- Such an interaction is supported in the command line execution of programs

```
#include <simplecpp>
main_program{
    turtleSim();
    cout << "No. of sides?";
    int noofsides;
    cin >> noofsides;
    repeat(noofsides)
        { forward(10);
          right(360.0/
            noofsides);
        }
}
```



Thank You

Happy Prutoring!