

Frequently Asked Questions

1. **What is “Parikshak”?**

Parikshak is an Automated Program Grading & Analysis Tool. It allows examiners to conduct programming exams in online mode, auto-evaluates candidates’ programming assignments, gives intuitive feedback to candidates, and does qualitative analysis of candidates’ programs after the programming test.

2. **What programming languages are supported in Parikshak for the SEBI exam?**

Parikshak supports the following 3 programming languages and versions.

Language	Version
C++ (GCC)	9.3
Java	1.8
Python	3.5

3. **Does Parikshak use its own compiler?**

No, Parikshak uses the standard compiler and version for each language. See Answer No. 2 for languages and versions.

4. **What are the different types of problems that will be given to the candidates during SEBI exams?**

There are two types of problems that are given to the candidates:

1. The first type of problem requires the development of a complete program where candidates have to write the program from start to end. This includes importing libraries, reading inputs, processing data, and printing output.
2. In the second type of problem, candidates just have to complete the function/routine and return the value; they do not have to read any data or print any value. The input parameters are already provided to the candidates; they just need to process the input and return the output.

5. **What is the duration of the exam?**

The total duration of the exam will be 3.5 hours with the first half hour for working on a paper (without computers) and the next 3 hours for coding on computers. For the first 30 minutes of the test, typing in Parikshak will be blocked.

6. **What is an attempt? Can I get more attempts?**

An attempt is one session of the exam duration; for the SEBI exam, there will be only one attempt of 3 hours.

Note: there will be no extra attempt once **you** close the exam.

7. **When can we close the exam?**

(i) Once you have solved all the problems; i.e got all “Y” for each problem that you needed to solve, or (ii) you are no longer interested to continue the exam. However, before closing the exam please contact the SEBI official available at the exam venue.

Note: Once you close the exam, you will not be allowed to log in again.

8. **What will happen if the attempt submission duration is over to solve the problems?**

Once the attempt is over, the system will be closed automatically, after which no submission will be allowed for any of the problems. The evaluation will be considered only for those solutions for which “**Submit to Grader**” was done during the attempt duration.

Note: During the exam, no extra attempt will be given.

9. **How can I test/debug my program before submitting my program for evaluation to Parikshak?**

You can provide your own test case in the input box “**Provide your own test cases:**” and click the “**Self Assessment**” button.

This will first compile your program and then run the program against your inputs. You can see the output of your program in the “**output**” box along with all debug statements.

Note: Remove or /*comment out */ all the debug statements before “Submit to Grader” for evaluation.

10. **What will happen when a program goes on an infinite loop in Parikshak?**

There is no way to find a program that is in an infinite loop, it's known as a classical problem which cannot be solved. So in Parikshak, a program is killed explicitly when a problem takes more time and memories than the predefined time and memories for a particular problem.

11. **What will be the different outputs that I get during the “Self Assessment”?**

- a. **Compilation error:** When the code has any syntax error. *Below example of C++ code has a syntax error, the system will produce the syntax error as:*

C++ Program “Test.cpp”	Parikshak Output
<pre> 1. #include <iostream> 2. using namespace std; 3. int main(){ 4. double n; 5. cin>>n; 6. int i = 0; 7. while(n > 0){ 8. n = n/2; 9. i++; 10. } 11. If(i > 0) 12. cout<<i<<endl; 13. return 0; 14. }</pre>	<p>Test.cpp: In function 'int main()':</p> <p>Test.cpp:4:10: error: 'n' was not declared in this scope</p> <pre> 4 cin>>n; </pre>

- b. **No output:** When the code does not produce any output. *In the below example the value of n is not provided or the input is “0” for “n”.*

C++ Program “Test.cpp”	Parikshak Output
<pre> 1. #include <iostream> 2. using namespace std; 3. int main(){ 4. double n; 5. cin>>n;</pre>	<p>Execution Time : 0.00 Second</p> <p>Execution Space : 4 Kbyte</p> <p>-----</p> <p>[no-output]</p>

<pre> 6. int i = 0; 7. while(n > 0){ 8. n = n/2; 9. i++; 10. } 11. If(i > 0) 12. cout<<i<<endl; 13. return 0; 14. }</pre>	
---	--

- c. **Correct output:** When the code will produce the output without any errors as expected. In the below example if the value of *i* is greater than 0 then it is printed.

C++ Program "Test.cpp"	Parikshak Output
<pre> 1. #include <iostream> 2. using namespace std; 3. int main(){ 4. double n; 5. cin>>n; 6. int i = 0; 7. while(n > 0){ 8. n = n/2; 9. i++; 10. } 11. If(i > 0) 12. cout<<i<<endl; 13. return 0; 14. }</pre>	<p>Execution Time: 0.00 Second Execution Space : 4 Kbyte ----- i = 1078</p>

- d. **"Time limit or memory limit exceeded"** When the program is in an infinite loop or taking more time or memory than expected. In the below example for any value of "n" greater than "0", the program will go into an infinite loop.

C++ Program "Test.cpp"	Parikshak Output
<pre> 1. #include <iostream></pre>	<p>Execution Time: 0.00 Second</p>

<pre> 2. using namespace std; 3. int main(){ 4. double n; 5. cin>>n; 6. int i = 0; 7. while(n >= 0){ 8. n = n/2; 9. i++; 10. } 11. If(i > 0) 12. cout<<i<<endl; 13. return 0; 14. } </pre>	<p>Execution Space: 4 Kbyte</p> <p>-----</p> <p>Time limit or Memory limit exceeded</p>
---	---

12. What is the purpose of “Submit to Grader”?

This is very important for your program to be considered for evaluations. For every program that you want to be evaluated, you need to do **“Submit to Grader”**. If you fail to do it, that program will not be considered for evaluations and no marks will be awarded irrespective of whether your program is correct or not.

13. What does Parikshak do when we do “Submit to Grader”? What is the grader's output?

When a program is submitted to the grader for a selected problem; the grader tests the program against the hidden test cases based on the problem statement and input specification of the problem and checks the output as per the output specification. For every passed (output as expected) test case, you will get a “Y” and for failed (output not as expected) test cases you will get an “X”. The final grade is a sequence of “Y” and “X” for each program and each submission. Once you get all “Y”, your program is accepted against all the test cases and you can move to the next problem.

Note: In case your program has syntax error or giving runtime error, even then the grader will give only **“X”** for each test case.

14. How many times can I do “Submit to Grader” for one program?

There is no restriction on the no of times a candidate can perform “Submit to Grader” in one attempt of 3 hours.

15. What is the input specification of the problem?

This specifies the sequence of the input data; type of data (i.e integer, character, float, string, etc.) and meaning of each input. Note that, do not print anything on the screen while reading the input such as “Enter the Number:”, “Number of Rows:” etc.

16. What is the importance of the output specification of the problem?

It is very important to follow the output specification. You should not print/produce any output other than the output asked in the specification.

Example: If the output specification has been asked for the sum of numbers:

- a. The output should be only the number without any other character before or after i.e. “90”
- b. If you print the “sum of the number is 90” instead of just “90”, your program will be evaluated wrong as “X” during grader evaluation.

17. What are low, Medium and High complexity problems?

- a. **Low complexity:** The problem will be based on basic operations on Array, String, Matrix etc.
- b. **Medium complexity:** The problem will be based on data structures; such as stack, queue, linked-list, trees, graphs, etc.
- c. **High Complexity:** It will be algorithms based problems; such as sorting, searching, optimization, etc.

18. How many problems will be given during the exam?

Please refer to the following table:

Complexity	Maximum Question	Needed to be solved	Maximum Max Per Question
Low	4	3	10
Medium	3	2	20
High	2	1	30

19. Is there an order for solving the questions?

No order needs to be followed; however, it is advisable to solve simple problems first and then move to the complex ones.

20. Is there any restriction on the libraries in the languages for the SEBI exam?

There is no restriction from Parikshak for using any standard library available with the language, however, no third-party library is available or allowed in Parikshak which is not part of the standard library in a particular version of the language.

21. What should be the filename of my solutions in Parikshak?

Your filename can be anything for all the languages, except the “**Java**” language. For java files, you have to give the same name of the class, the class which contains the **main** function.

All the filenames should have appropriate extensions as given below:

Language	File extension
Java	.java
C++	.cpp
Python	.py

22. Can we create multiple files for a problem?

No, you just need to create only one file for a problem. All the classes and functions should be written inside one file only, for a single problem.

23. Any Disclaimer about Parikshak Software?

Yes.

Disclaimer: Parikshak program grading is purely input/output based. SEBI/test conducting body reserves the right to manually examine the submitted code against malpractices and reported errors related to problem specifications. When there is adequate ground, SEBI/test conducting body may override Parikshak's verdict by granting/denying credit.
