

PROGRAMMING PRACTICES MINI PROJECT

AKSHAY KUMAR VERMA

October 29, 2023

1 Introduction

This project serves as one stop destination for all your needs related for competitive examinations. The current scenario in competitive world shows that the competition is increased too much whether it is in the field of JEE,NEET,Clat or it is about any higher engineering examinations like GATE.The AIM of this interface is to cater the candidates with all possible and the best resources for the respective competitive examinations they are going to phase.

2 Methodology

The project is made up using the cpp programming language.This report also includes the other things as well like the *profiling* and *debugging* operations perfomed on the code and the respective results obtained.

2.1 Flow of code

The flow of the code goes in the way that:

1. There is a prompt for the category to choose for.
2. On the basis of chosen category **viewBookdetails** function is called. Which displays the different sub available int categories.
3. Then prompt is there to enter integer digit to choose which subject need to choose.
4. Then on the basis of integer input the **viewSubDetails** function is called.
5. Then it list all the best book resouces to follow for that subject.
- 6.Then further either you can exit the terminal by entering 0 or can check for any other subject as well.
- 7.This is how the code goes...

3 Scalability

This interface is a basic one but this can be scale to large scale by adding many more features such as adding tests,youtube resources, teaching assistants,doubt resolver and many more facilities like that...

4 Debugging

The process of identifying and removing errors from code is debugging. Here is how the debugging process was applied on the code and the results i got:

1: It shows that compilation was not successfull due to d was not declared in scope and also not defined properly

Compilation failed due to following error(s).

```
main.cpp: In function 'int main()':
main.cpp:381:13: error: 'd' was not declared in this scope
  381 |             d=choice - 1;
      |             ^
```

(a) Image 1

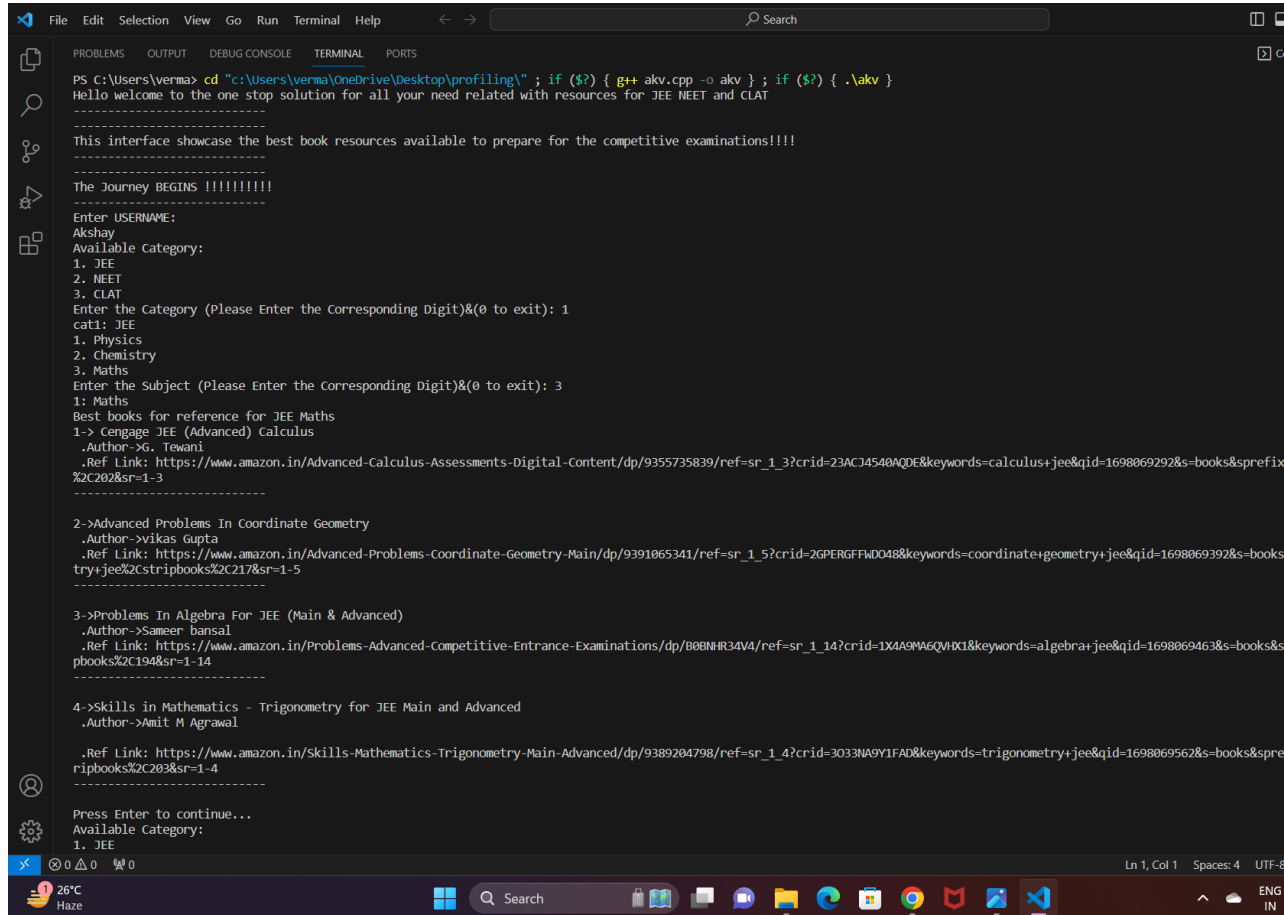
2: It shows on debugging the d was declared in the another scope and on entering value of d the result getting was not the expected one.

```
371             cout << "Enter the Category (Please Enter the Corresponding Digit)&(0 to exit): ";
(gdb) n
372             cin >> choice;
(gdb) n
Enter the Category (Please Enter the Corresponding Digit)&(0 to exit): 2
374             if (choice < 1 || choice > bookResources.size())
(gdb) n
383             viewBookDetails(bookResources[d],d); // calls the view book details
function
(gdb) n
cat1: JEE
1. Physics
2. Chemistry
3. Maths
```

Figure 2: Image 2

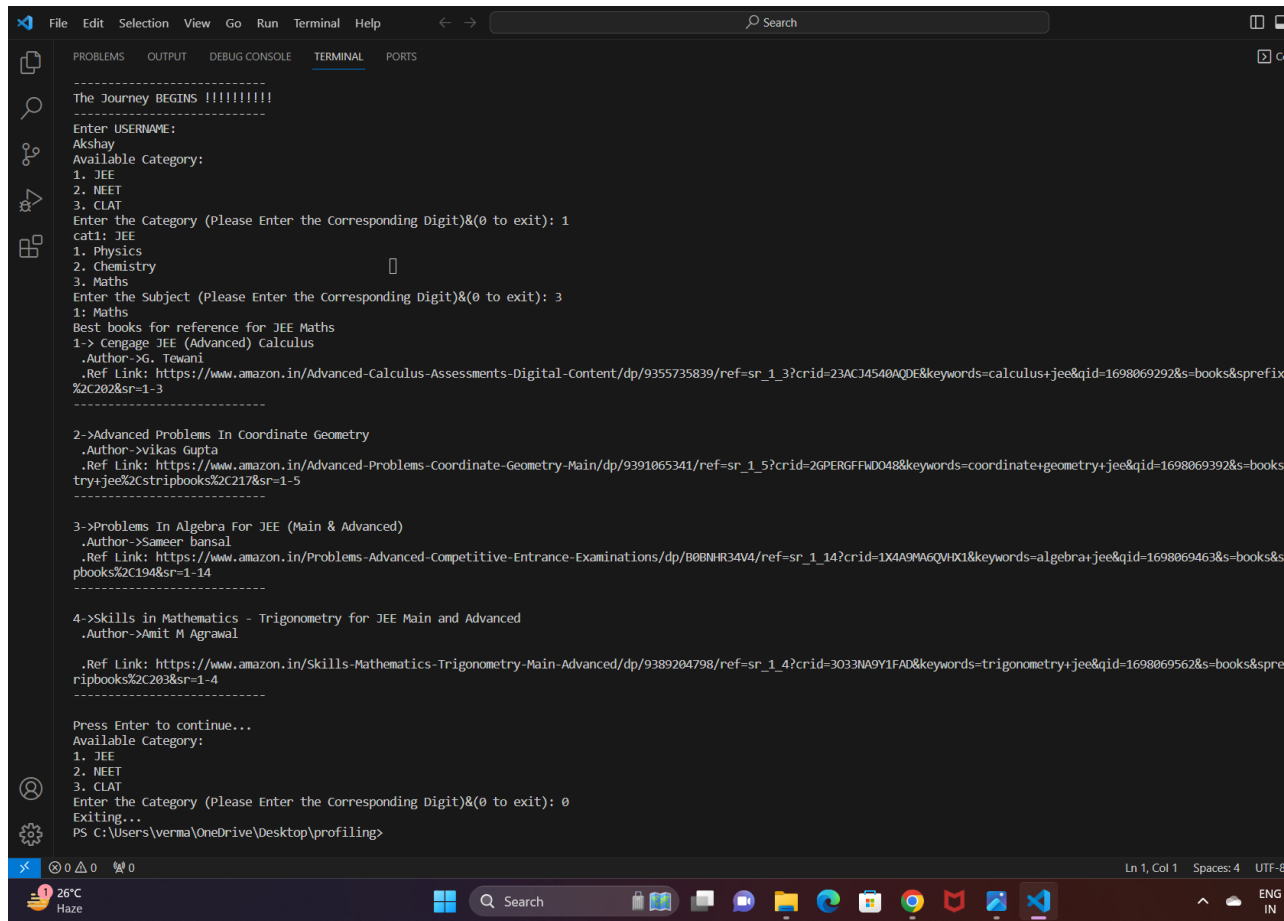
5 Output

The generated output on running the code:



```
PS C:\Users\verma> cd "c:\Users\verma\OneDrive\Desktop\profiling\"; if ($?) { g++ akv.cpp -o akv }; if ($?) { .\akv }
Hello welcome to the one stop solution for all your need related with resources for JEE NEET and CLAT
-----
This interface showcase the best book resources available to prepare for the competitive examinations!!!!
-----
The Journey BEGINS !!!!!!!!!!!
-----
Enter USERNAME:
Akshay
Available Category:
1. JEE
2. NEET
3. CLAT
Enter the Category (Please Enter the Corresponding Digit)&(0 to exit): 1
cat1: JEE
1. Physics
2. Chemistry
3. Maths
Enter the Subject (Please Enter the Corresponding Digit)&(0 to exit): 3
1: Maths
Best books for reference for JEE Maths
1-> Cengage JEE (Advanced) Calculus
.Author->G. Tewani
.Ref Link: https://www.amazon.in/Advanced-Calculus-Assessments-Digital-Content/dp/9355735839/ref=sr_1_3?crid=23ACJ4540AQDE&keywords=calculus+jee&qid=1698069292&s=books&prefix=
%2C202&sr=1-3
-----
2->Advanced Problems In Coordinate Geometry
.Author->vikas Gupta
.Ref Link: https://www.amazon.in/Advanced-Problems-Coordinate-Geometry-Main/dp/9391065341/ref=sr_1_5?crid=2GPERGFFWD048&keywords=coordinate+geometry+jee&qid=1698069392&s=books&
try+jee%2Cstripbooks%2C217&sr=1-5
-----
3->Problems In Algebra For JEE (Main & Advanced)
.Author->Sameer bansal
.Ref Link: https://www.amazon.in/Problems-Advanced-Competitive-Entrance-Examinations/dp/B08BNR34V4/ref=sr_1_14?crid=1X4A9MA6QVHX1&keywords=algebra+jee&qid=1698069463&s=books&
pbooks%2C194&sr=1-14
-----
4->Skills in Mathematics - Trigonometry for JEE Main and Advanced
.Author->Amit M Agrawal
.Ref Link: https://www.amazon.in/Skills-Mathematics-Trigonometry-Main-Advanced/dp/9389204798/ref=sr_1_4?crid=3033NA9Y1FAD&keywords=trigonometry+jee&qid=1698069562&s=books&sp
ripbooks%2C203&sr=1-4
-----
Press Enter to continue...
Available Category:
1. JEE
```

Figure 3: Image 11



```
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

The Journey BEGINS !!!!!!!!

Enter USERNAME:
Akshay
Available Category:
1. JEE
2. NEET
3. CLAT
Enter the Category (Please Enter the Corresponding Digit)&(0 to exit): 1
cat1: JEE
1. Physics
2. Chemistry
3. Maths
Enter the Subject (Please Enter the Corresponding Digit)&(0 to exit): 3
1: Maths
Best books for reference for JEE Maths
1-> Cengage JEE (Advanced) Calculus
.Author->G. Tewani
.Ref Link: https://www.amazon.in/Advanced-Calculus-Assessments-Digital-Content/dp/9355735839/ref=sr_1_3?crd=23ACJ4540AQDE&keywords=calculus+jee&qid=1698069292&s=books&prefix=calculus&pf_rd_p=1698069292&pf_rd_r=1698069292&sr=1-3
-----
2->Advanced Problems In Coordinate Geometry
.Author->vikas Gupta
.Ref Link: https://www.amazon.in/Advanced-Problems-Coordinate-Geometry-Main/dp/9391065341/ref=sr_1_5?crd=2GPERGFFWD048&keywords=coordinate+geometry+jee&qid=1698069392&s=books&pf_rd_p=1698069392&pf_rd_r=1698069392&sr=1-5
-----
3->Problems In Algebra For JEE (Main & Advanced)
.Author->Sameer bansal
.Ref Link: https://www.amazon.in/Problems-Advanced-Competitive-Entrance-Examinations/dp/B08BNHR34V4/ref=sr_1_14?crd=1X4A9MA6QV4X1&keywords=algebra+jee&qid=1698069463&s=books&pf_rd_p=1698069463&pf_rd_r=1698069463&sr=1-14
-----
4->Skills in Mathematics - Trigonometry for JEE Main and Advanced
.Author->Amit M Agrawal
.Ref Link: https://www.amazon.in/Skills-Mathematics-Trigonometry-Main-Advanced/dp/9389204798/ref=sr_1_4?crd=3033NA9Y1FA0&keywords=trigonometry+jee&qid=1698069562&s=books&pf_rd_p=1698069562&pf_rd_r=1698069562&sr=1-4
-----
Press Enter to continue...
Available Category:
1. JEE
2. NEET
3. CLAT
Enter the Category (Please Enter the Corresponding Digit)&(0 to exit): 0
Exiting...
PS C:\Users\verma\OneDrive\Desktop\profiling>
```

Figure 4: Image 12

6 Profiling

On profiling using different *gprof* commands got the following flat profile and call graph:

Windows PowerShell

Flat profile:

Each sample counts as 0.01 seconds.
no time accumulated

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
0.00	0.00	0.00	60	0.00	0.00	std::vector<Subject, std::allocator<Subject> >::size() const
0.00	0.00	0.00	45	0.00	0.00	Subject&& std::forward<Subject>(std::remove_reference<Subject>::type&)
0.00	0.00	0.00	36	0.00	0.00	std::move_iterator<Subject*>::base() const
0.00	0.00	0.00	34	0.00	0.00	std::vector<Category, std::allocator<Category> >::size() const
0.00	0.00	0.00	27	0.00	0.00	Subject::~Subject()
0.00	0.00	0.00	27	0.00	0.00	__gnu_cxx::new_allocator<Subject>::max_size() const
0.00	0.00	0.00	27	0.00	0.00	Subject* std::_addressof<Subject>(Subject&)
0.00	0.00	0.00	24	0.00	0.00	operator new(unsigned int, void*)
0.00	0.00	0.00	22	0.00	0.00	std::_Vector_base<Subject, std::allocator<Subject> >::_M_get_Tp_allocator()
0.00	0.00	0.00	18	0.00	0.00	Subject::Subject(Subject&&)
0.00	0.00	0.00	18	0.00	0.00	std::_Vector_base<Subject, std::allocator<Subject> >::_M_get_Tp_allocator() const
0.00	0.00	0.00	18	0.00	0.00	std::vector<Subject, std::allocator<Subject> >::max_size() const
0.00	0.00	0.00	18	0.00	0.00	std::move_iterator<Subject*>::move_iterator(Subject*)
0.00	0.00	0.00	18	0.00	0.00	std::allocator_traits<std::allocator<Subject> >::max_size(std::allocator<Subject> co
0.00	0.00	0.00	18	0.00	0.00	std::move_iterator<Subject*> std::_make_move_if_noexcept_iterator<Subject, std::mov
>						>(Subject)
0.00	0.00	0.00	18	0.00	0.00	void std::_Destroy<Subject>(Subject*)
0.00	0.00	0.00	18	0.00	0.00	bool std::operator==(Subject*)(std::move_iterator<Subject*> const&, std::move_iterat
&)						
0.00	0.00	0.00	18	0.00	0.00	bool std::operator!=(Subject*)(std::move_iterator<Subject*> const&, std::move_iterat
&)						
0.00	0.00	0.00	15	0.00	0.00	Category&& std::forward<Category>(std::remove_reference<Category>::type&)
0.00	0.00	0.00	13	0.00	0.00	void std::_Destroy_aux<false>::_destroy<Subject*>(Subject*, Subject*)
0.00	0.00	0.00	13	0.00	0.00	std::_Vector_base<Subject, std::allocator<Subject> >::_M_deallocate(Subject*, unsign
0.00	0.00	0.00	13	0.00	0.00	void std::_Destroy<Subject*>(Subject*, Subject*)
0.00	0.00	0.00	13	0.00	0.00	void std::_Destroy<Subject*, Subject>(Subject*, Subject*, std::allocator<Subject>&)
0.00	0.00	0.00	12	0.00	0.00	std::move_iterator<Category*>::base() const
0.00	0.00	0.00	12	0.00	0.00	std::vector<Category, std::allocator<Category> >::operator[](unsigned int) const
0.00	0.00	0.00	12	0.00	0.00	unsigned int const& std::max<unsigned int>(unsigned int const&, unsigned int const&)
0.00	0.00	0.00	9	0.00	0.00	Category::~Category()
0.00	0.00	0.00	9	0.00	0.00	__gnu_cxx::new_allocator<Subject>::deallocate(Subject*, unsigned int)
0.00	0.00	0.00	9	0.00	0.00	__gnu_cxx::new_allocator<Subject>::allocate(unsigned int, void const*)
0.00	0.00	0.00	9	0.00	0.00	void __gnu_cxx::new_allocator<Subject>::construct<Subject, Subject>(Subject*, Subject
0.00	0.00	0.00	9	0.00	0.00	__gnu_cxx::new_allocator<Category>::max_size() const
0.00	0.00	0.00	9	0.00	0.00	std::move_iterator<Subject*>::operator*() const
0.00	0.00	0.00	9	0.00	0.00	std::vector<Subject, std::allocator<Subject> >::_M_check_len(unsigned int, char cons
0.00	0.00	0.00	9	0.00	0.00	std::vector<Subject, std::allocator<Subject> >::operator[](unsigned int) const
0.00	0.00	0.00	9	0.00	0.00	std::_Vector_base<Subject, std::allocator<Subject> >::_M_allocate(unsigned int)

32°C Sunny

Search

ENG IN

Figure 5: Image 3

```

0.00 0.00 0.00 9 0.00 0.00 std::_Vector_base<Subject, std::allocator<Subject> >::_M_allocate(unsigned int)
0.00 0.00 0.00 9 0.00 0.00 std::move_iterator<Subject*>::operator++()
0.00 0.00 0.00 9 0.00 0.00 std::allocator_traits<std::allocator<Subject> >::_M_deallocate(std::allocator<Subject>&,
ed int)
0.00 0.00 0.00 9 0.00 0.00 std::allocator_traits<std::allocator<Subject> >::_M_allocate(std::allocator<Subject>&,
0.00 0.00 0.00 9 0.00 0.00 void std::allocator_traits<std::allocator<Subject> >::_construct<Subject, Subject>(std:
ct>&, Subject*, Subject&&)
0.00 0.00 0.00 9 0.00 0.00 Subject* std::__uninitialized_copy<false>::__uninit_copy<std::move_iterator<Subject*>,
move_iterator<Subject*>, std::move_iterator<Subject*>, Subject*)
0.00 0.00 0.00 9 0.00 0.00 void std::vector<Subject, std::allocator<Subject> >::_M_emplace_back<Subject>(Subject&&
0.00 0.00 0.00 9 0.00 0.00 void std::vector<Subject, std::allocator<Subject> >::_M_emplace_back_aux<Subject>(Sub
0.00 0.00 0.00 9 0.00 0.00 std::vector<Subject, std::allocator<Subject> >::_push_back(Subject&&)
0.00 0.00 0.00 9 0.00 0.00 void std::_Construct<Subject, Subject>(Subject*, Subject&&)
0.00 0.00 0.00 9 0.00 0.00 Category* std::_addressof<Category>(Category&)
0.00 0.00 0.00 9 0.00 0.00 Subject* std::__uninitialized_copy<std::move_iterator<Subject*>, Subject*>(std::move_i
std::move_iterator<Subject*>, Subject*)
0.00 0.00 0.00 9 0.00 0.00 Subject* std::__uninitialized_copy_a<std::move_iterator<Subject*>, Subject*, Subject*
or<Subject*>, std::move_iterator<Subject*>, Subject*,
0.00 0.00 0.00 9 0.00 0.00 Subject* std::__uninitialized_move_if_noexcept_a<Subject*, Subject*, std::allocator<
*, Subject*, Subject*, std::allocator<Subject>&)
0.00 0.00 0.00 9 0.00 0.00 std::remove_reference<Subject&>::type&& std::move<Subject&>(Subject&)
0.00 0.00 0.00 7 0.00 0.00 std::_Vector_base<Category, std::allocator<Category> >::_M_get_Tp_allocator()
0.00 0.00 0.00 6 0.00 0.00 Category::Category(Category&&)
0.00 0.00 0.00 6 0.00 0.00 std::_Vector_base<Category, std::allocator<Category> >::_M_get_Tp_allocator() const
0.00 0.00 0.00 6 0.00 0.00 std::vector<Category, std::allocator<Category> >::max_size() const 0.00 0.00
0.00 0.00 std::move_iterator<Category*>::move_iterator(Category*)
0.00 0.00 0.00 6 0.00 0.00 std::allocator_traits<std::allocator<Category> >::max_size(std::allocator<Category>
0.00 0.00 0.00 6 0.00 0.00 std::move_iterator<Category*> std::__make_move_if_noexcept_iterator<Category, std::m
ory*> >(Category*)
0.00 0.00 0.00 6 0.00 0.00 void std::_Destroy<Category>(Category*)
0.00 0.00 0.00 6 0.00 0.00 bool std::operator==<Category*>(std::move_iterator<Category*> const&, std::move_iter
nst&)
0.00 0.00 0.00 6 0.00 0.00 bool std::operator!=<Category*>(std::move_iterator<Category*> const&, std::move_iter
nst&)
0.00 0.00 0.00 4 0.00 0.00 displayBookResources(std::vector<Category, std::allocator<Category> > const&)
0.00 0.00 0.00 4 0.00 0.00 __gnu_cxx::new_allocator<Subject>::new_allocator()
0.00 0.00 0.00 4 0.00 0.00 __gnu_cxx::new_allocator<Subject>::~new_allocator()
0.00 0.00 0.00 4 0.00 0.00 std::allocator<Subject>::allocator()
0.00 0.00 0.00 4 0.00 0.00 std::allocator<Subject>::~allocator()
0.00 0.00 0.00 4 0.00 0.00 void std::_Destroy_aux<false>::_destroy<Category*>(Category*, Category*)
0.00 0.00 0.00 4 0.00 0.00 std::_Vector_base<Subject, std::allocator<Subject> >::_Vector_impl::_Vector_impl()
0.00 0.00 0.00 4 0.00 0.00 std::_Vector_base<Subject, std::allocator<Subject> >::_Vector_impl::~_Vector_impl()
0.00 0.00 0.00 4 0.00 0.00 std::_Vector_base<Subject, std::allocator<Subject> >::_Vector_base()
0.00 0.00 0.00 4 0.00 0.00 std::_Vector_base<Subject, std::allocator<Subject> >::~_Vector_base()
0.00 0.00 0.00 4 0.00 0.00 std::_Vector_base<Category, std::allocator<Category> >::_M_deallocate(Category*, uns

```

Figure 6: Image 4

```
Windows PowerShell
0.00 0.00 0.00 4 0.00 0.00 std::_Vector_base<Category, std::allocator<Category> >::_M_deallocate(Category*, uns
0.00 0.00 0.00 4 0.00 0.00 std::vector<Subject, std::allocator<Subject> >::vector()
0.00 0.00 0.00 4 0.00 0.00 std::vector<Subject, std::allocator<Subject> >::~~vector()
0.00 0.00 0.00 4 0.00 0.00 void std::_Destroy<Category*>(Category*, Category*)
0.00 0.00 0.00 4 0.00 0.00 void std::_Destroy<Category*, Category>(Category*, Category*, std::allocator<Category
0.00 0.00 0.00 3 0.00 0.00 displaySubjects(std::vector<Subject, std::allocator<Subject> > const&)
0.00 0.00 0.00 3 0.00 0.00 viewBookDetails(Category&, int)
0.00 0.00 0.00 3 0.00 0.00 __gnu_cxx::new_allocator<Category>::deallocate(Category*, unsigned int)
0.00 0.00 0.00 3 0.00 0.00 __gnu_cxx::new_allocator<Category>::allocate(unsigned int, void const*)
0.00 0.00 0.00 3 0.00 0.00 void __gnu_cxx::new_allocator<Category>::construct<Category, Category>(Category*, Ca
0.00 0.00 0.00 3 0.00 0.00 std::move_iterator<Category*>::operator*() const
0.00 0.00 0.00 3 0.00 0.00 std::vector<Category, std::allocator<Category> >::_M_check_len(unsigned int, char co
0.00 0.00 0.00 3 0.00 0.00 std::_Vector_base<Category, std::allocator<Category> >::_M_allocate(unsigned int)
0.00 0.00 0.00 3 0.00 0.00 std::move_iterator<Category*>::operator++()
0.00 0.00 0.00 3 0.00 0.00 std::allocator_traits<std::allocator<Category> >::deallocate(std::allocator<Category
igned int)
0.00 0.00 0.00 3 0.00 0.00 std::allocator_traits<std::allocator<Category> >::allocate(std::allocator<Category>&
0.00 0.00 0.00 3 0.00 0.00 void std::allocator_traits<std::allocator<Category> >::construct<Category, Category>
tegrity>&, Category*, Category&&)
0.00 0.00 0.00 3 0.00 0.00 Category* std::::_uninitialized_copy<false>::__uninit_copy<std::move_iterator<Category
d::move_iterator<Category*>, std::move_iterator<Category*>, Category*)
0.00 0.00 0.00 3 0.00 0.00 std::vector<Subject, std::allocator<Subject> >::operator[](unsigned int)
0.00 0.00 0.00 3 0.00 0.00 void std::vector<Category, std::allocator<Category> >::emplace_back<Category>(Catego
0.00 0.00 0.00 3 0.00 0.00 void std::vector<Category, std::allocator<Category> >::_M_emplace_back_aux<Category>
0.00 0.00 0.00 3 0.00 0.00 std::vector<Category, std::allocator<Category> >::push_back(Category&&)
0.00 0.00 0.00 3 0.00 0.00 std::vector<Category, std::allocator<Category> >::operator[](unsigned int)
0.00 0.00 0.00 3 0.00 0.00 void std::_Construct<Category, Category>(Category*, Category&&)
0.00 0.00 0.00 3 0.00 0.00 Category* std::::_uninitialized_copy<std::move_iterator<Category*>, Category*>(std::mov
y*>, std::move_iterator<Category*>, Category*)
0.00 0.00 0.00 3 0.00 0.00 Category* std::::_uninitialized_copy_a<std::move_iterator<Category*>, Category*, Cate
erator<Category*>, std::move_iterator<Category*>, Category*, std::allocator<Category>&)
0.00 0.00 0.00 3 0.00 0.00 Category* std::::_uninitialized_move_if_noexcept_a<Category*, Category*, std::allocat
egory*, Category*, Category*, std::allocator<Category>&)
0.00 0.00 0.00 3 0.00 0.00 std::remove_reference<Category&>::type&& std::move<Category&>(Category&)
0.00 0.00 0.00 1 0.00 0.00 viewSubDetailsC(Subject&, int)
0.00 0.00 0.00 1 0.00 0.00 viewSubDetailsJ(Subject&, int)
0.00 0.00 0.00 1 0.00 0.00 viewSubDetailsN(Subject&, int)
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<Category>::new_allocator()
0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::new_allocator<Category>::~new_allocator()
0.00 0.00 0.00 1 0.00 0.00 std::allocator<Category>::allocator()
0.00 0.00 0.00 1 0.00 0.00 std::allocator<Category>::~allocator()
0.00 0.00 0.00 1 0.00 0.00 std::_Vector_base<Category, std::allocator<Category> >::_Vector_impl::_Vector_impl()
0.00 0.00 0.00 1 0.00 0.00 std::_Vector_base<Category, std::allocator<Category> >::_Vector_impl::~~Vector_impl()
0.00 0.00 0.00 1 0.00 0.00 std::_Vector_base<Category, std::allocator<Category> >::_Vector_base()
0.00 0.00 0.00 1 0.00 0.00 std::_Vector_base<Category, std::allocator<Category> >::~~Vector_base()
```

Figure 7: Image 5

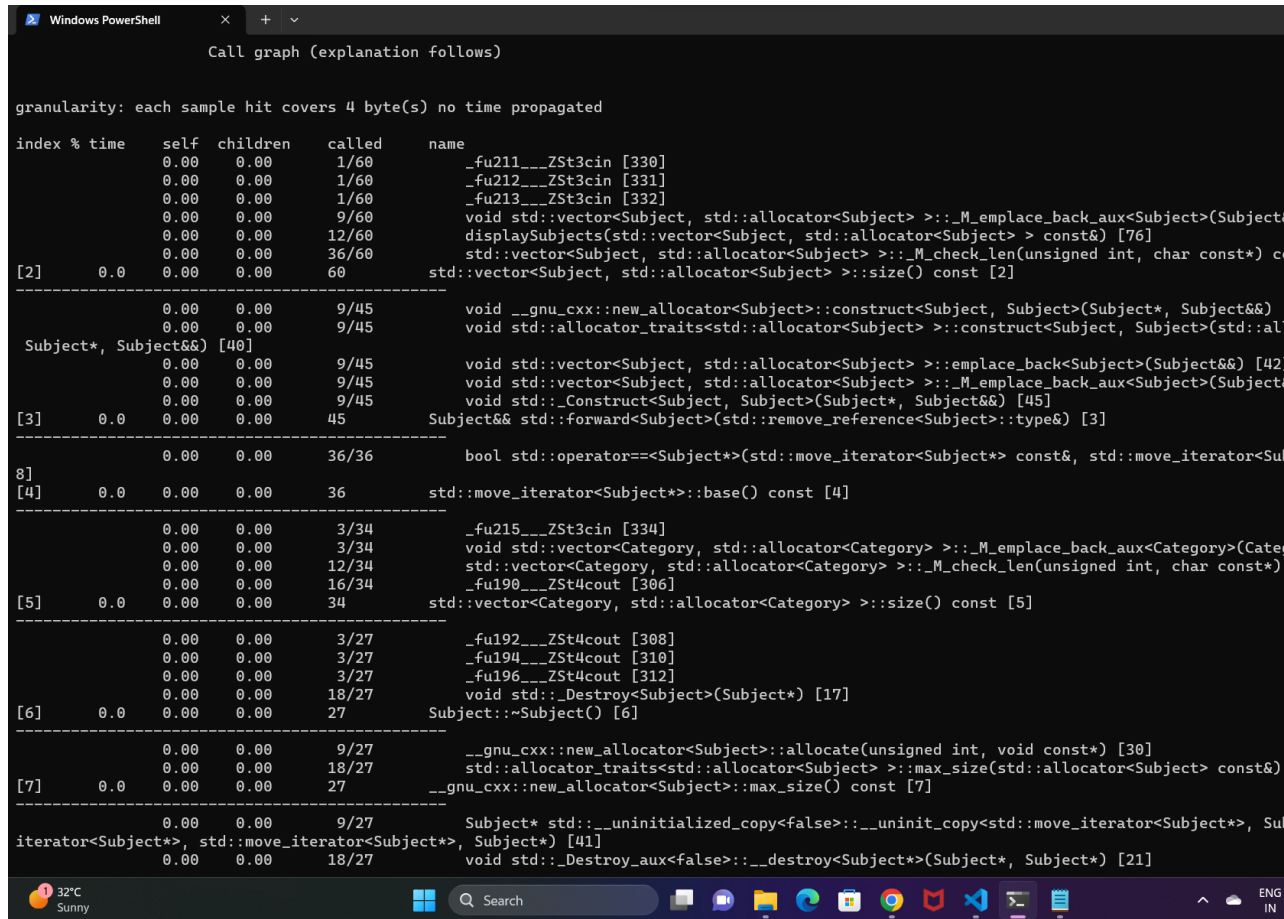


Figure 8: Image 6

```

Windows PowerShell
[8] 0.0 0.00 0.00 18/27 void std::_Destroy_aux<false>::_destroy<Subject*>(Subject*, Subject*) [21]
0.0 0.00 0.00 27 Subject* std::_addressof<Subject>(Subject&) [8]
-----
0.00 0.00 3/24 void __gnu_cxx::new_allocator<Category>::construct<Category, Category>(Category*, Category*) [94]
0.00 0.00 3/24 void std::_Construct<Category, Category>(Category*, Category&&) [94]
0.00 0.00 9/24 void __gnu_cxx::new_allocator<Subject>::construct<Subject, Subject>(Subject*, Subject&&) [45]
0.00 0.00 9/24 void std::_Construct<Subject, Subject>(Subject*, Subject&&) [45]
[9] 0.0 0.00 0.00 24 operator new(unsigned int, void*) [9]
-----
0.00 0.00 4/22 std::vector<Subject, std::allocator<Subject> >::~vector() [73]
0.00 0.00 18/22 void std::vector<Subject, std::allocator<Subject> >::_M_emplace_back_aux<Subject>(Subject*) [10]
[10] 0.0 0.00 0.00 22 std::_Vector_base<Subject, std::allocator<Subject> >::_M_get_Tp_allocator() [10]
-----
0.00 0.00 9/18 void __gnu_cxx::new_allocator<Subject>::construct<Subject, Subject>(Subject*, Subject&&) [45]
0.00 0.00 9/18 void std::_Construct<Subject, Subject>(Subject*, Subject&&) [45]
[11] 0.0 0.00 0.00 18 Subject::Subject(Subject&&) [11]
-----
0.00 0.00 18/18 std::vector<Subject, std::allocator<Subject> >::max_size() const [13]
[12] 0.0 0.00 0.00 18 std::_Vector_base<Subject, std::allocator<Subject> >::_M_get_Tp_allocator() const [12]
-----
0.00 0.00 18/18 std::vector<Subject, std::allocator<Subject> >::_M_check_len(unsigned int, char const*) const [13]
[13] 0.0 0.00 0.00 18 std::vector<Subject, std::allocator<Subject> >::max_size() const [13]
0.00 0.00 18/18 std::_Vector_base<Subject, std::allocator<Subject> >::_M_get_Tp_allocator() const [12]
0.00 0.00 18/18 std::allocator_traits<std::allocator<Subject> >::max_size(std::allocator<Subject> const&) [15]
-----
0.00 0.00 18/18 std::move_iterator<Subject*> std::_make_move_if_noexcept_iterator<Subject, std::move_iterator<Subject*> [16]
[14] 0.0 0.00 0.00 18 std::move_iterator<Subject*>::move_iterator(Subject*) [14]
-----
0.00 0.00 18/18 std::vector<Subject, std::allocator<Subject> >::max_size() const [13]
[15] 0.0 0.00 0.00 18 std::allocator_traits<std::allocator<Subject> >::max_size(std::allocator<Subject> const&) [15]
0.00 0.00 18/27 __gnu_cxx::new_allocator<Subject>::max_size() const [7]
-----
0.00 0.00 18/18 Subject* std::_uninitialized_move_if_noexcept_a<Subject*, Subject*, std::allocator<Subject> [49]
[16] 0.0 0.00 0.00 18 std::move_iterator<Subject*> std::_make_move_if_noexcept_iterator<Subject, std::move_iterator<Subject*> [16]
-----
0.00 0.00 18/18 std::move_iterator<Subject*>::move_iterator(Subject*) [14]
-----
0.00 0.00 18/18 void std::_Destroy_aux<false>::_destroy<Subject*>(Subject*, Subject*) [21]
[17] 0.0 0.00 0.00 18 void std::_Destroy<Subject>(Subject*) [17]
0.00 0.00 18/27 Subject::~Subject() [6]
-----
0.00 0.00 18/18 bool std::operator!=<Subject*>(std::move_iterator<Subject*> const&, std::move_iterator<Subject*> [9]

```

Figure 9: Image 7

```
Windows PowerShell
iterator<Subject*>, std::move_iterator<Subject*>, Subject*) [41]
[19] 0.0 0.00 0.00 18 bool std::operator!=<Subject*>(std::move_iterator<Subject*> const&, std::move_iterator<Subject*>
0.00 0.00 0.00 18/18 bool std::operator==<Subject*>(std::move_iterator<Subject*> const&, std::move_iterator<Subject*>
8]
-----
0.00 0.00 3/15 void __gnu_cxx::new_allocator<Category>::construct<Category, Category>(Category*, Category
0.00 0.00 3/15 void std::allocator_traits<std::allocator<Category> >::construct<Category, Category>(std:
y>&, Category*, Category&&) [87]
0.00 0.00 3/15 void std::vector<Category, std::allocator<Category> >::emplace_back<Category>(Category&&)
0.00 0.00 3/15 void std::vector<Category, std::allocator<Category> >::_M_emplace_back_aux<Category>(Cate
0.00 0.00 3/15 void std::_Construct<Category, Category>(Category*, Category&&) [94]
[20] 0.0 0.00 0.00 15 Category&& std::forward<Category>(std::remove_reference<Category>::type&) [20]
-----
0.00 0.00 13/13 void std::_Destroy<Subject*>(Subject*, Subject*) [23]
[21] 0.0 0.00 0.00 13 void std::_Destroy_aux<false>::_destroy<Subject*>(Subject*, Subject*) [21]
/27 Subject* std::_addressof<Subject>(Subject&) [8]
0.00 0.00 18/18 void std::_Destroy<Subject>(Subject*) [17]
-----
0.00 0.00 4/13 std::_Vector_base<Subject, std::allocator<Subject> >::~_Vector_base() [70]
0.00 0.00 9/13 void std::vector<Subject, std::allocator<Subject> >::_M_emplace_back_aux<Subject>(Subject
[22] 0.0 0.00 0.00 13 std::_Vector_base<Subject, std::allocator<Subject> >::_M_deallocate<Subject*, unsigned int> [
0.00 0.00 9/9 std::allocator_traits<std::allocator<Subject> >::deallocate(std::allocator<Subject>&, Sub
t) [38]
-----
0.00 0.00 13/13 void std::_Destroy<Subject*, Subject>(Subject*, Subject*, std::allocator<Subject>&) [24]
[23] 0.0 0.00 0.00 13 void std::_Destroy<Subject*>(Subject*, Subject*) [23]
0.00 0.00 13/13 void std::_Destroy_aux<false>::_destroy<Subject*>(Subject*, Subject*) [21]
-----
0.00 0.00 4/13 std::vector<Subject, std::allocator<Subject> >::~vector() [73]
0.00 0.00 9/13 void std::vector<Subject, std::allocator<Subject> >::_M_emplace_back_aux<Subject>(Subject
[24] 0.0 0.00 0.00 13 void std::_Destroy<Subject*, Subject>(Subject*, Subject*, std::allocator<Subject>&) [24]
0.00 0.00 13/13 void std::_Destroy<Subject*>(Subject*, Subject*) [23]
-----
0.00 0.00 12/12 bool std::operator==<Category*>(std::move_iterator<Category*> const&, std::move_iterator<
[59]
[25] 0.0 0.00 0.00 12 std::move_iterator<Category*>::base() const [25]
-----
0.00 0.00 12/12 _fu190___ZSt4cout [306]
[26] 0.0 0.00 0.00 12 std::vector<Category, std::allocator<Category> >::operator[](unsigned int) const [26]
-----
0.00 0.00 3/12 std::vector<Category, std::allocator<Category> >::_M_check_len(unsigned int, char const*)
0.00 0.00 9/12 std::vector<Subject, std::allocator<Subject> >::_M_check_len(unsigned int, char const*) c
[27] 0.0 0.00 0.00 12 unsigned int const& std::max<unsigned int>(unsigned int const&, unsigned int const&) [27]
-----
0.00 0.00 3/9 _fu214___ZSt3cin [333]
```

Figure 10: Image 8

```

Windows PowerShell
[49] 0.0 0.00 0.00 9/9 void std::vector<Subject, std::allocator<Subject> >::_M_emplace_back_aux<Subject>(Subject
t*, Subject*, std::allocator<Subject>&) [49] Subject* std::_uninitialized_move_if_noexcept_a<Subject*, Subject*, std::allocator<Subject>
0.00 0.00 18/18 std::move_iterator<Subject*> std::_make_move_if_noexcept_iterator<Subject, std::move_ite
Subject*) [16] 0.00 0.00 9/9 Subject* std::_uninitialized_copy_a<std::move_iterator<Subject*>, Subject*, Subject>(std
bject*>, std::move_iterator<Subject*>, Subject*, std::allocator<Subject>&) [48]
-----
[50] 0.0 0.00 0.00 9/9 std::vector<Subject, std::allocator<Subject> >::push_back(Subject&&) [44]
std::remove_reference<Subject&&>::type&& std::move<Subject&&>(Subject&) [50]
-----
[51] 0.0 0.00 0.00 1/7 std::vector<Category, std::allocator<Category> >::~vector() [111]
0.00 0.00 6/7 void std::vector<Category, std::allocator<Category> >::_M_emplace_back_aux<Category>(Cate
std::_Vector_base<Category, std::allocator<Category> >::_M_get_Tp_allocator() [51]
-----
[52] 0.0 0.00 0.00 3/6 void __gnu_cxx::new_allocator<Category>::construct<Category, Category>(Category*, Category
0.00 0.00 3/6 void std::_Construct<Category, Category>(Category*, Category&&) [94]
Category::Category(Category&&) [52]
-----
std::vector<Category, std::allocator<Category> >::max_size() const [54][53] 0.0 0.0
std::_Vector_base<Category, std::allocator<Category> >::_M_get_Tp_allocator() const [53]
-----
[54] 0.0 0.00 0.00 6/6 std::vector<Category, std::allocator<Category> >::_M_check_len(unsigned int, char const*)
0.00 0.00 6/6 std::vector<Category, std::allocator<Category> >::max_size() const [54]
0.00 0.00 6/6 std::_Vector_base<Category, std::allocator<Category> >::_M_get_Tp_allocator() const [53]
0.00 0.00 6/6 std::allocator_traits<std::allocator<Category> >::max_size(std::allocator<Category> const
-----
>(Category*) [57] std::move_iterator<Category*> std::_make_move_if_noexcept_iterator<Category, std::move_i
[55] 0.0 0.00 0.00 6/6 std::move_iterator<Category*>::move_iterator(Category*) [55]
-----
std::vector<Category, std::allocator<Category> >::max_size() const [54][56] 0.0 0.0
std::allocator_traits<std::allocator<Category> >::max_size(std::allocator<Category> const&) [56]
0.00 0.00 6/9 __gnu_cxx::new_allocator<Category>::max_size() const [32]
-----
Category* std::_uninitialized_move_if_noexcept_a<Category*, Category*, std::allocator<Ca
*, Category*, Category*, std::allocator<Category>&) [97]
[57] 0.0 0.00 0.00 6/6 std::move_iterator<Category*> std::_make_move_if_noexcept_iterator<Category, std::move_ite
ategory*) [57]
0.00 0.00 6/6 std::move_iterator<Category*>::move_iterator(Category*) [55]
-----
[58] 0.0 0.00 0.00 6/6 void std::_Destroy_aux<false>::_destroy<Category*>(Category*, Category*) [66]
0.00 0.00 6/6 void std::_Destroy<Category>(Category*) [58]
0.00 0.00 6/9 Category::~Category() [28]
-----

```

Figure 11: Image 9

