

Assignment #07
0801CS211008
BREAKOUT

Objective Of Project :

Python is widely used by game development companies and for creating mobile games. We will be creating a simple ball breakout game using it.

Breakout is one of the earliest arcade video games. This one-player game features simple 2D graphics. It consists of one paddle used to return a bouncing ball back and forth across the screen. The aim of the game is to break the bricks of a brick wall by getting the ball to hit/bounce on the bricks. The score correspond to the number of bricks being hit.

Statiscal information :

Starting date : 13, November, 2022

End Date : 18, November, 2022

Total Time Required : 5 days

Total Line Of Code (Python): 497

Number Of Funtions (Python) : 29

Function description (Python) :

File : main.py

1. pause_game : This function is used to pause the game at any moment during playing by using a global game_pause boolean.
2. stop : This function is used to end the game at any moment.
3. check_coll_walls : This function is used to check the collision of ball with any of the walls by using the distance between coordinates of the ball's center and the walls and then if collides ball is bounced in the opposite direction. If ball collides with the bottom wall one life is lost.
4. check_coll_paddle : This function is used to check the collision of ball with the paddle by using the distance between coordinates of the ball's center and the paddle's center and then if collides ball is bounced in the opposite direction.
5. check_coll_brick : This function is used to check the collision of ball with the bricks by using the distance between coordinates of the ball's center and the bricks' center and then if collides ball is bounced in the opposite direction and that brick is removed.

File : ball.py

6. left : This function is used to move the ball backwards.
7. right : This function is used to move the ball forwards.
8. move : To move the ball.
9. bounce : When the ball collides with any surface this function is used to make it bounce or move in another direction.
10. reset : To reset the position of the ball.
11. init : It is called implicitly to do the basic tasks/creations.

File : brick.py

12. init : It is called implicitly to do the basic tasks/creations.
13. crt_lane : To create a lane of bricks.
14. crt_lanes : To create and join all the lanes of bricks.

15. reset : To reset the arrangement of bricks to their original form.

File : pdle.py

16. init : It is called implicitly to do the basic tasks/creations.

17. left : This function is used to move the paddle backwards.

18. right : This function is used to move the paddle forwards.

19. reset : To reset the position of the paddle.

File : scrbrd.py

20. init : It is called implicitly to do the basic tasks/creations.

21. update_score : Used to update the current score/lives/high score and we write it on the screen.

22. inc_score : when collision with a brick this function is used to increase the current score.

23. dec_lvs : used to decrease a life when ball collides with bottom wall.

24. reset : to reset the score to 0.

File : ui.py

25. init : It is called implicitly to do the basic tasks/creations.

26. GO : to show win/lose status when game over.

27. when_pause : shows current score when the game is paused.

28. unpauses : used to resume the game.

29. header : a heading is shown in the center with this function.

Profile Report (Python) :

main.py

Microsoft Windows [Version 10.0.19044.2251]

(c) Microsoft Corporation. All rights reserved.

C:\Users\91826\projects\Games and projects\ball breakout game py>python -m cProfile main.py > main prof.txt

C:\Users\91826\projects\Games and projects\ball breakout game py>python -m cProfile main.py
21198831 function calls (21198359 primitive calls) in 67.205 seconds

Ordered by: standard name

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
18	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:100(acquire)
18/6	0.000	0.000	0.051	0.009	<frozen importlib._bootstrap>:1022(_find_and_load)
8/2	0.000	0.000	0.007	0.003	<frozen importlib._bootstrap>:1053(_handle_fromlist)
18	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:125(release)
18	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:165(__init__)
18	0.000	0.000	0.001	0.000	<frozen importlib._bootstrap>:169(__enter__)
18	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:173(__exit__)
18	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:179(_get_module_lock)
18	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:198(cb)
25/6	0.000	0.000	0.043	0.007	<frozen importlib._bootstrap>:233(_call_with_frames_remo
154	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:244(_verbose_message)
4	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:254(_requires_builtin_wrap
18	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:357(__init__)
27	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:391(cached)
18	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:404(parent)
18	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:412(has_location)

```

4 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:421(spec_from_loader)
13 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:48(new_module)
18 0.000 0.000 0.001 0.000 <frozen importlib._bootstrap>:492(_init_module_attrs)
18 0.000 0.000 0.016 0.001 <frozen importlib._bootstrap>:564(module_from_spec)
18/6 0.000 0.000 0.049 0.008 <frozen importlib._bootstrap>:664(_load_unlocked)
18 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:71(__init__)
18 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:746(find_spec)
4 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:770(create_module)
4 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:778(exec_module)
4 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:795(is_package)
14 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:826(find_spec)
46 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:893(_enter_)
46 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:897(_exit_)
18 0.000 0.000 0.005 0.000 <frozen importlib._bootstrap>:921(_find_spec)
18/6 0.000 0.000 0.051 0.008 <frozen importlib._bootstrap>:987(_find_and_load_unlocke
13 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap_external>:1040(__init__)
13 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap_external>:1065(get_filename
13 0.000 0.000 0.012 0.001 <frozen importlib._bootstrap_external>:1070(get_data)

```

ball.py

C:\Users\91826\projects\Games and projects\ball_breakout_game_py>python -m cProfile ball.py > ball_prof.txt

C:\Users\91826\projects\Games and projects\ball_breakout_game_py>python -m cProfile ball.py
7826 function calls (7560 primitive calls) in 0.039 seconds

Ordered by: standard name

```

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:100(acquire)
8/1    0.000    0.000    0.039    0.039 <frozen importlib._bootstrap>:1022(_find_and_load)
6/1    0.000    0.000    0.002    0.002 <frozen importlib._bootstrap>:1053(_handle_fromlist)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:125(release)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:165(__init__)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:169(_enter_)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:173(_exit_)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:179(_get_module_lock)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:198(cb)
12/1   0.000    0.000    0.035    0.035 <frozen importlib._bootstrap>:233(_call_with_frames_removed)
79     0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:244(_verbose_message)
1      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:254(_requires_builtin_wrapper)

1      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:421(spec_from_loader)
6      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:48(new_module)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:492(_init_module_attrs)
8      0.000    0.000    0.014    0.002 <frozen importlib._bootstrap>:564(module_from_spec)
8/1    0.000    0.000    0.038    0.038 <frozen importlib._bootstrap>:664(_load_unlocked)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:71(__init__)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:746(find_spec)
1      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:770(create_module)
1      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:778(exec_module)
1      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:795(is_package)
7      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:826(find_spec)
22     0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:893(_enter_)
22     0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:897(_exit_)
8      0.000    0.000    0.003    0.000 <frozen importlib._bootstrap>:921(_find_spec)

```

bricks.py

C:\Users\91826\projects\Games and projects\ball_breakout_game_py>python -m cProfile bricks.py > bricks_prof.txt

```
C:\Users\91826\projects\Games and projects\ball_breakout_game_py>python -m cProfile bricks.py
7828 function calls (7562 primitive calls) in 0.040 seconds
```

Ordered by: standard name

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:100(acquire)
8/1	0.000	0.000	0.040	0.040	<frozen importlib._bootstrap>:1022(_find_and_load)
6/1	0.000	0.000	0.004	0.004	<frozen importlib._bootstrap>:1053(_handle_fromlist)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:125(release)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:165(__init__)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:169(__enter__)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:173(__exit__)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:179(_get_module_lock)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:198(cb)
12/1	0.000	0.000	0.039	0.039	<frozen importlib._bootstrap>:233(_call_with_frames_removed)
79	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:244(_verbose_message)
1	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:254(_requires_builtin_wrapper)
8/1	0.000	0.000	0.040	0.040	<frozen importlib._bootstrap>:664(_load_unlocked)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:71(__init__)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:746(find_spec)
1	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:770(create module)
1	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:778(exec_module)
1	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:795(is_package)
7	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:826(find_spec)
22	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:893(__enter__)
22	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:897(__exit__)
8	0.000	0.000	0.003	0.000	<frozen importlib._bootstrap>:921(_find_spec)
8/1	0.000	0.000	0.040	0.040	<frozen importlib._bootstrap>:987(_find_and_load_unlocked)
6	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap_external>:1040(__init__)
6	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap_external>:1065(get_filename)
6	0.000	0.000	0.001	0.000	<frozen importlib._bootstrap_external>:1070(get_data)
6	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap_external>:1089(path_stats)

pdle.py

```
C:\Users\91826\projects\Games and projects\ball_breakout_game_py>python -m cProfile pdle.py > pdle_prof.txt
```

```
C:\Users\91826\projects\Games and projects\ball_breakout_game_py>python -m cProfile pdle.py
7826 function calls (7560 primitive calls) in 0.047 seconds
```

Ordered by: standard name

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:100(acquire)
8/1	0.000	0.000	0.047	0.047	<frozen importlib._bootstrap>:1022(_find_and_load)
6/1	0.000	0.000	0.003	0.003	<frozen importlib._bootstrap>:1053(_handle_fromlist)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:125(release)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:165(__init__)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:169(__enter__)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:173(__exit__)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:179(_get_module_lock)
8	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:198(cb)
12/1	0.000	0.000	0.044	0.044	<frozen importlib._bootstrap>:233(_call_with_frames_removed)
79	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:244(_verbose_message)
1	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:254(_requires_builtin_wrapper)

```

8 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:404(parent)
8 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:412(has_location)
1 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:421(spec_from_loader)
6 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:48(_new_module)
8 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:492(_init_module_attrs)
8 0.000 0.000 0.019 0.002 <frozen importlib._bootstrap>:564(module_from_spec)
8/1 0.000 0.000 0.046 0.046 <frozen importlib._bootstrap>:664(_load_unlocked)
8 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:71(__init__)
8 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:746(find_spec)
1 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:770(create_module)
1 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:778(exec_module)
1 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:795(is_package)
7 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:826(find_spec)
22 0.000 0.000 0.000 0.000 <frozen importlib._bootstrap>:893(__enter__)

```

scrbrd.py

```
C:\Users\91826\projects\Games and projects\ball_breakout_game_py>python -m cProfile scrbrd.py > scrbrd_prof.txt
```

```
C:\Users\91826\projects\Games and projects\ball_breakout_game_py>python -m cProfile scrbrd.py
7831 function calls (7565 primitive calls) in 0.043 seconds
```

Ordered by: standard name

```

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:100(acquire)
8/1    0.000    0.000    0.042    0.042 <frozen importlib._bootstrap>:1022(_find_and_load)
6/1    0.000    0.000    0.002    0.002 <frozen importlib._bootstrap>:1053(_handle_fromlist)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:125(release)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:165(__init__)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:169(__enter__)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:173(__exit__)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:179(_get_module_lock)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:198(cb)

8      0.000    0.000    0.019    0.002 <frozen importlib._bootstrap>:564(module_from_spec)
8/1    0.000    0.000    0.041    0.041 <frozen importlib._bootstrap>:664(_load_unlocked)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:71(__init__)
8      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:746(find_spec)
1      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:770(create_module)
1      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:778(exec_module)
1      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:795(is_package)
7      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:826(find_spec)
22     0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:893(__enter__)
22     0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:897(__exit__)
8      0.000    0.000    0.004    0.000 <frozen importlib._bootstrap>:921(_find_spec)
8/1    0.000    0.000    0.042    0.042 <frozen importlib._bootstrap>:987(_find_and_load_unlocked)
6      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:1040(__init__)
6      0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:1065(get_filename)
6      0.000    0.000    0.002    0.000 <frozen importlib._bootstrap_external>:1070(get_data)

```

ui.py

```
C:\Users\91826\projects\Games and projects\ball_breakout_game_py>python -m cProfile ui.py > ui_prof.txt
```

```
C:\Users\91826\projects\Games and projects\ball_breakout_game_py>python -m cProfile ui.py
8963 function calls (8675 primitive calls) in 0.038 seconds
```

Ordered by: standard name

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
13	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:100(acquire)
13/2	0.000	0.000	0.038	0.019	<frozen importlib._bootstrap>:1022(_find_and_load)
6/1	0.000	0.000	0.002	0.002	<frozen importlib._bootstrap>:1053(_handle_fromlist)
13	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:125(release)
13	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:165(_init_)
13	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:169(_enter_)
13	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:173(_exit_)
13	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:179(_get_module_lock)
13	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:198(cb)
20/2	0.000	0.000	0.035	0.018	<frozen importlib._bootstrap>:233(_call_with_frames_removed)
124	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:244(_verbose_message)
13	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:492(_init_module_attrs)
13	0.000	0.000	0.014	0.001	<frozen importlib._bootstrap>:564(module_from_spec)
13/2	0.000	0.000	0.037	0.018	<frozen importlib._bootstrap>:664(_load_unlocked)
13	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:71(_init_)
13	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:746(find_spec)
4	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:770(create_module)
4	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:778(exec_module)
4	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:795(is_package)
9	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:826(find_spec)
31	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:893(_enter_)
31	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:897(_exit_)
13	0.000	0.000	0.003	0.000	<frozen importlib._bootstrap>:921(_find_spec)

Debug (Python) :

```
C:\Users\91826\projects\Games and projects\ball_breakout_game_py>python -m pdb main.py
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(4)<module>()
-> import turtle as t
(Pdb) run
Restarting C:\Users\91826\projects\Games and projects\ball_breakout_game_py\main.py with arguments:

> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(4)<module>()
-> import turtle as t
(Pdb) n
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(5)<module>()
-> import time
(Pdb) n
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(9)<module>()
-> from paddle import Paddle
(Pdb) break 21
*** Blank or comment
(Pdb) break 25
Breakpoint 1 at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:25
(Pdb) break 35
Breakpoint 2 at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:35
(Pdb) break 52
Breakpoint 3 at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:52
(Pdb) break 74
Breakpoint 4 at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:74
```



```

(Pdb) break
Num Type      Disp Enb  Where
1  breakpoint keep yes   at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:25
2  breakpoint keep yes   at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:35
3  breakpoint keep yes   at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:52
4  breakpoint keep yes   at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:74
(Pdb) disable 2
Disabled breakpoint 2 at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:35
(Pdb) break
Num Type      Disp Enb  Where
1  breakpoint keep yes   at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:25
2  breakpoint keep no    at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:35
3  breakpoint keep yes   at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:52
4  breakpoint keep yes   at c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py:74
(Pdb) jump 47
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(47)<module>()
-> game_pause = False
(Pdb) n
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(48)<module>()
-> play_game = True
(Pdb) whatis game_pause
<class 'bool'>
(Pdb) n
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(51)<module>()
-> def pause_game():
(Pdb) n
(Pdb) where
c:\users\91826\appdata\local\programs\python\python310\lib\bdb.py(597)run()
-> exec(cmd, globals, locals)
<string>(1)<module>()
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(61)<module>()
-> def stop():
(Pdb) step
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(66)<module>()
-> scr.listen()
-> check_coll_brick()
(Pdb) n
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(217)<module>()
-> if len(bricks.brks) == 0:
(Pdb) step
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(193)<module>()
-> while play_game:
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(193)<module>()
-> while play_game:
(Pdb) n
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(196)<module>()
-> if not game_pause:
(Pdb)
> c:\users\91826\projects\games and projects\ball_breakout_game_py\main.py(198)<module>()
-> if TEMP:
(Pdb)
(Pdb) help

Documented commands (type help <topic>):
=====
EOF      c          d          h          list       q          rv          undisplay
a        cl         debug      help       ll          quit       s          unt
alias    clear      disable    ignore     longlist   r          source     until
args     commands  display    interact   n          restart    step       up
b        condition down       j          next       return     tbreak     w
break    cont       enable     jump       p          retval    u          whatis
bt       continue  exit       l          pp         run        unalias    where

Miscellaneous help topics:
=====
exec  pdb

```

Code (Python) :

```
main.py  
# main.py for game background working  
  
# turtle module for using console window  
import turtle as t  
import time  
  
# importing classes  
# PADDLE  
from pdle import Paddle  
# UI  
from ui import UI  
# BRICKS  
from bricks import Bricks  
# scoreboard  
from scrbrd import Scoreboard  
# BALL  
from ball import Ball  
  
TEMP = False  
  
# creating console background  
scr = t.Screen()  
scr.setup(width=1200,height=600)  
scr.bgcolor('black')  
scr.title('BREAKOUT')  
scr.tracer(0)  
  
# instances of classes  
# PADDLE  
pd=Paddle()  
# UI  
ui = UI()  
# BRICKS  
bricks = Bricks()  
# scoreboard  
score = Scoreboard(lives = 3)  
# BALL  
bl=Ball()  
  
# calling header function to write heading  
ui.header()
```



```
# initialising game
game_pause = False
play_game = True

# function to pause the game
def pause_game():
    global game_pause, ui
    if game_pause:
        # ui.header()
        game_pause = False
    else:
        game_pause = True

# function to end game
def stop():
    global play_game
    play_game = False

scr.listen()

# defining which keys it needs to listen to,
# which are the left and right arrow keys (to
# move the paddle) s to stop and R to restart.
scr.onkey(key='Left', fun = pd.left)
scr.onkey(key='Right', fun = pd.right)
scr.onkey(key='s', fun = stop)
scr.onkey(key='space', fun = pause_game)

# CHECKING THE COLLISIONS OF BALL

# WALLS
def check_coll_walls():

    global bl, score, play_game, ui

    # collision with left and right walls
    if bl.xcor() < -580 or bl.xcor() > 570:
        bl.bounce(x_bounce=True, y_bounce=False)
        return

    # collision with upper wall
    if bl.ycor() > 240 :
        bl.bounce(x_bounce=False, y_bounce=True)
```

```

        return

    # collision with lower wall,
    # in which case user fails and game resets
    if bl.ycor() < -280:
        bl.reset()
        score.dec_lvs()

    # GAME OVER and uses lost if lives = 0
    if score.lvs == 0:
        score.reset()
        ui.GO(W=False)
        time.sleep(2)
        play_game = False
        return
    return

# PADDLE
def check_coll_paddle():

    global bl, pd

    # x coordinates of paddle and ball
    pd_x = pd.xcor()
    bl_x = bl.xcor()

    # checking for coll
    # by using centre of ball and paddle
    if bl.distance(pd) < 110 and bl.ycor() < -250 :

        # when paddle is right of scr
        if pd_x > 0:
            if bl_x > pd_x:
                # balls hits left side
                bl.bounce(x_bounce=True, y_bounce=True)
                return

            else:
                bl.bounce(x_bounce=False, y_bounce=True)
                return

        # when paddle is left of scr
        elif pd_x < 0 :
            if bl_x < pd_x:
                # balls hits left side
                bl.bounce(x_bounce=True, y_bounce=True)
                return

```

```

        else:
            bl.bounce(x_bounce=False, y_bounce=True)
            return

# paddle in centre
    else:
        if bl.x == pd.x:
            bl.bounce(x_bounce=False, y_bounce=True)
            return
        else:
            bl.bounce(x_bounce=True, y_bounce=True)
            return

# BRICKS
def check_coll_brick():

    global bl, bricks, score

    for brick in bricks.brks:

        # when ball in range
        if bl.distance(brick) < 40:

            brick.clear()
            brick.goto(3000,3000)
            bricks.brks.remove(brick)

            # detection coll to bounce the ball
            # LEFT
            if bl.xcor() < brick.left_wall:
                bl.bounce(x_bounce=True, y_bounce=False)

            # RIGHT
            elif bl.xcor() > brick.right_wall:
                bl.bounce(x_bounce=True, y_bounce=False)

            # BOTTOM
            elif bl.ycor() < brick.bottom_wall:
                bl.bounce(x_bounce=False, y_bounce=True)

            # TOP
            elif bl.ycor() > brick.upper_wall:
                bl.bounce(x_bounce=False, y_bounce=True)

            # increasing score
            score.inc_score()

```

```
time.sleep(1)
# starting the game
while play_game:

    # playing if game not paused
    if not game_pause:

        if TEMP:
            TEMP = False
            ui.header()

        # updating console
        scr.update()
        time.sleep(0.02)
        bl.move()

        # checking collisions with walls
        check_coll_walls()

        # checking collision with paddle
        check_coll_paddle()

        # checking collisions with bricks
        check_coll_brick()

        # checking for win situation
        if len(bricks.brks) == 0:
            ui.GO(W=True)
            break

    else:
        TEMP = True
        ui.when_pause(score.score)

scr.bye()
t.mainloop()

bricks.py

# bricks.py to create and control appearance and
# disappearance of bricks in the game

# turtle module to create the basic structure
from turtle import Turtle
```

```
# class Brick for one brick definition
# Brick class inheriting Turtle class
class Brick(Turtle):

    # function to define brick's properties
    def __init__(self, x_cor, y_cor):
        super().__init__()
        self.penup()

        # creating brick
        self.shape('square')
        self.shapesize(stretch_len=3, stretch_wid=1.5)

        # assigning them color
        self.color('red')
        self.goto(x=x_cor, y=y_cor)

        # border
        self.left_wall = self.xcor() - 30
        self.right_wall = self.xcor() + 30
        self.upper_wall = self.ycor() + 15
        self.bottom_wall = self.ycor() - 15

# class Bricks for definition of collection of bricks
class Bricks(Turtle):

    # function for the properties of collection
    def __init__(self) -> None:
        self.y_str = 0
        self.y_end = 240
        self.brks = []
        self.crt_lanes()

    # function for one lane of bricks
    def crt_lane(self, y_cor):
        for i in range(-570, 570, 63):
            brick = Brick(i, y_cor)
            self.brks.append(brick)

    # function for all the lanes
    def crt_lanes(self):
        for i in range(self.y_str, self.y_end, 32):
            self.crt_lane(i)

    # function to reset
    def reset(self):
        self.y_str = 0
        self.y_end = 240
```

```
        self.brks = []
        self.crt_lanes()
```

pdle.py

```
# pdle.py to create and control movements of paddle
# for the game
```

```
# turtle module to create the basic structure
from turtle import Turtle
```

```
# DIS to move for paddle in one movement
DIS = 70
```

```
# paddle class inheriting turtle class
class Paddle(Turtle):
```

```
    # function to define paddle properties
    def __init__(self):
        super().__init__()

        # creating paddle
        self.color('white')
        self.shape('square')
        self.penup()
        self.shapesize(stretch_len=10,stretch_wid=1)
        self.goto(x=0,y=-280)

    # left function moves paddle to the left
    def left(self):
        self.backward(DIS)

    # right function moves paddle to the left
    def right(self):
        self.forward(DIS)

    # function to reset
    def reset(self):
        self.goto(x=0,y=-280)
```

ball.py

```
# ball.py to create and control movements of ball
# for the game
```

```
# turtle module to create the basic structure
from turtle import Turtle
```

```
# DIS to move for ball in one movement
DIS = 10

# Ball class inheriting turtle class
class Ball(Turtle):

    # left function moves paddle to the left
    def left(self):
        self.backward(DIS)

    # right function moves paddle to the left
    def right(self):
        self.forward(DIS)

    # function to define ball's properties
    def __init__(self):
        super().__init__()

        # creating ball
        self.color('white')
        self.shape('circle')
        self.penup()

        # movement of ball each time
        self.x_move_dist=DIS
        self.y_move_dist=DIS

    # function to update the coordinates of ball
    def move(self):

        # moving 10 units in horizontal and
        # vertical directions
        new_y=self.ycor() + self.y_move_dist
        new_x=self.xcor() + self.x_move_dist
        self.goto(x=new_x,y=new_y)

    # function to bounce the ball in case of collision
    def bounce(self,x_bounce,y_bounce):

        # reversing direction horizontally
        if x_bounce:
            self.x_move_dist*=-1

        # reversing direction vertically
        if y_bounce:
            self.y_move_dist*=-1

    # function to reset the ball's position in case
```



```

    # of losing a life
    def reset(self):
        self.goto(x=0,y=-240)
        self.y_move_dist = DIS
        self.x_move_dist = DIS

scrbrd.py

# scrbrd.py to keep track of scores via
# a text file of the game

# turtle module to create
from turtle import Turtle

# file containing high score
FILE = 'C:/Users/91826/projects/Games_and_projects/ball_breakout_game_py/HighScore.txt'

# checking for the high score in text file
# if score not found either creating file or
# not high score = 0
try:
    score = int(open(FILE, 'r').read())
except FileNotFoundError:
    score = open(FILE, 'w').write(str(0))
except ValueError:
    score = 0

FONT = ('calibri',18,'normal')

# class Scoreboard for scoreboard defintions
# inheriting Turtle class
class Scoreboard(Turtle):

    # functon for scoreboard properties
    def __init__(self, lives) -> None:
        super().__init__()
        self.color('white')
        self.penup()
        self.hideturtle()

        # writing high score
        self.hs = score
        self.goto(x=-580,y=260)

        self.lvs=lives
        self.score = 0

        # updating score

```

```

        self.update_scr()

# function to update score
def update_scr(self):
    self.clear()
    self.write(f"Score: {self.score} | High Score: {self.hs} | Lives: {self.lvs}")

# increasing score
def inc_score(self):
    self.score += 1

    # if greater than high score then updating high score
    if self.score > self.hs :
        self.hs += 1
    self.update_scr()

# decreasing lives
def dec_lvs(self):
    self.lvs -= 1
    self.update_scr()

# when game ends resetting scores
def reset(self) -> None:
    self.clear()
    self.score = 0
    self.update_scr()
    open(FILE, 'w').write(str(self.hs))

```

ui.py

```

# ui.py to write methods required for user interface of the game

# importing turtle class from turtle module
from turtle import Turtle
import time, random

FONT = ('calibri', 52, 'normal')
FONT2 = ('calibri', 14, 'normal')
ALIGN = 'center'

# class UI inheriting Turtle
class UI(Turtle):

    # function for turtle properties to write
    def __init__(self) -> None:
        super().__init__()
        self.hideturtle()

```

```

        self.penup()
        # self.header()

# function for the heading
def header(self):
    self.clear()
    self.color('blue')
    self.goto(x = 0,y=-150 )
    self.write('BREAKOUT', align=ALIGN, font=FONT)
    self.goto(x=0,y=-180)
    self.write('Press Space to PAUSE or RESUME the Game',
               align=ALIGN, font=FONT2)

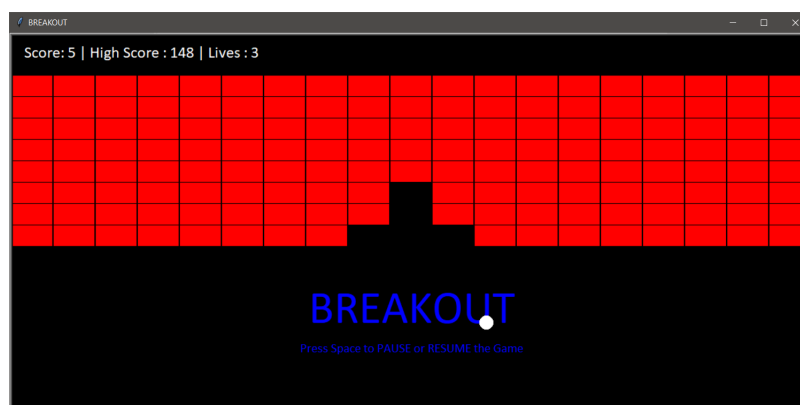
# function to show win/lose after the game ends
def GO(self,W):
    self.clear()
    if W == True:
        self.write("You Cleared the Game", align=ALIGN, font=FONT)
    else:
        self.write("Game Over! :( ", align=ALIGN, font=FONT)

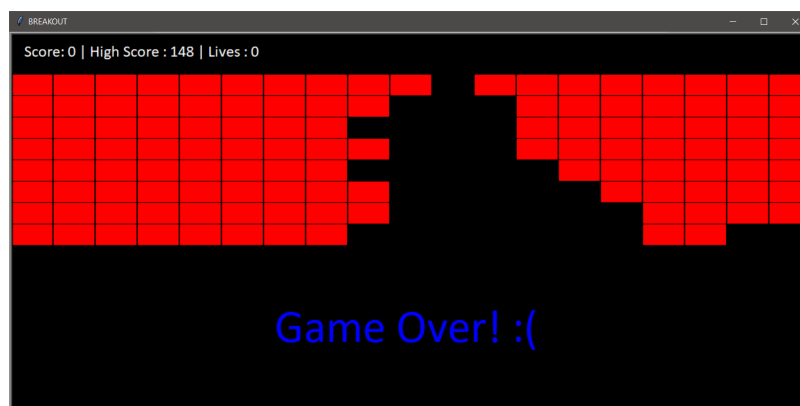
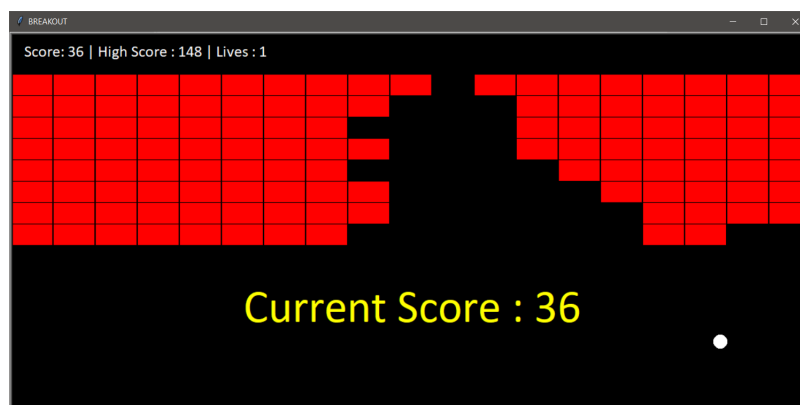
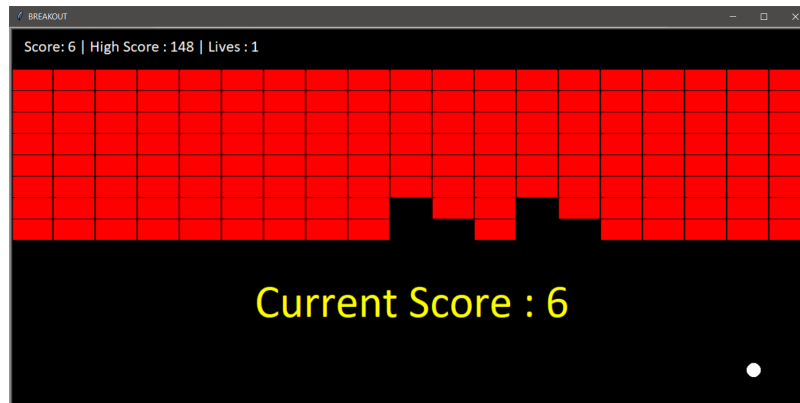
# function to show score when game is paused
def when_pause(self, score):
    self.clear()
    self.goto(x=0,y=-150)
    self.color('yellow')
    self.write(f"Current Score : {score}", align=ALIGN, font=FONT)

def unpause(self):
    self.clear()
    self.header()

```

Output (Python) :





SIMPLE CALCULATOR

Objective Of Project :

C++ is a cross-platform language that can be used to create high-performance applications. C++ is one of the world's most popular programming languages. C++ can be found in today's operating systems, Graphical User Interfaces, and embedded systems.

We will create a simple calculator using C++. This calculator will ask the user what operation they want to perform and then act accordingly and give the desired output.

Statiscal information :

Starting date : 17, November, 2022

End Date : 20, November, 2022

Total Time Required : 3 days

Total Line Of Code (C++): 267

Number Of Funtions (C++) : 17

Function description (C++) :

File : main.cpp

1. main : This is the main function used to begin the program.

File : interface.h

2. interface : This function is used to take the operations from user and check what it is.
3. check_op : This function checks the operation entered by user and then does the appropriate operation.

File : operations.h

4. add : This function performs addition of 2 given numbers.
5. sub : This function performs subtraction of 2 given numbers.
6. mul : This funtction multiplies 2 given numbers.
7. div : This function divides 2 given numbers.
8. percent : This functon finds num1 percent of num2. Where num1 and num2 are 2 given numbers.
9. fact : This function finds factorial of the given non-negative integer.
10. nPr : This function finds the permutations of 2 given numbers. Taking the bigger as n and smaller as r in nP_r .
11. nCr : This function finds the combinations of 2 given numbers. Taking the bigger as n and smaller as r in nC_r .
12. sqr : This function finds the square the given number.
13. cube : This function finds the cube of the given number.
14. power : This function finds num1 raised to the power num2. Both given by the user as input.
15. sine : This function finds sine of the given number considering to the be argument in radian.
16. cosine : This function finds cosine of the given number considering to the be argument in radian.
17. tangent : This function finds tangent of the given number considering to the be argument in radian.

Profile Report (C++) :

main.cpp

c:\Users\91826\projects\C\Calculator>g++ main.cpp -pg

c:\Users\91826\projects\C\Calculator>a.exe

Enter the operation you want to perform :

add

Enter the 2 numbers :

56 45

Sum : 101

Do you want to continue (y/n) : y

Enter the operation you want to perform :

54 69

SORRY!! I can't do this.

Do you want to continue (y/n) : Calculator closed!!

c:\Users\91826\projects\C\Calculator>gprof a.exe gmon.out

Flat profile:

Each sample counts as 0.01 seconds.
no time accumulated

% time	cumulative seconds	self seconds	self calls	self Ts/call	total Ts/call	name
-----------	-----------------------	-----------------	---------------	-----------------	------------------	------

% time	the percentage of the total running time of the program used by this function.					
-----------	--	--	--	--	--	--

cumulative seconds	a running sum of the number of seconds accounted for by this function and those listed above it.					
-----------------------	--	--	--	--	--	--

self seconds	the number of seconds accounted for by this function alone. This is the major sort for this listing.					
-----------------	--	--	--	--	--	--

calls	the number of times this function was invoked, if this function is profiled, else blank.					
-------	--	--	--	--	--	--

self	This is the total amount of time spent in this function.					
------	--	--	--	--	--	--

children	This is the total amount of time propagated into this function by its children.					
----------	---	--	--	--	--	--

called	This is the number of times the function was called. If the function called itself recursively, the number only includes non-recursive calls, and is followed by a '+' and the number of recursive calls.					
--------	---	--	--	--	--	--

name	The name of the current function. The index number is printed after it. If the function is a member of a cycle, the cycle number is printed between the function's name and the index number.					
------	---	--	--	--	--	--

self the average number of milliseconds spent in this
ms/call function per call, if this function is profiled,
 else blank.

total the average number of milliseconds spent in this
ms/call function and its descendents per call, if this
 function is profiled, else blank.

name the name of the function. This is the minor sort
 for this listing. The index shows the location of
 the function in the gprof listing. If the index is
 in parenthesis it shows where it would appear in
 the gprof listing if it were to be printed.

♀
Copyright (C) 2012-2021 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,
are permitted in any medium without royalty provided the copyright
notice and this notice are preserved.

♀

Call graph (explanation follows)

granularity: each sample hit covers 4 byte(s) no time propagated

index % time self children called name

This table describes the call tree of the program, and was sorted by
the total amount of time spent in each function and its children.

Each entry in this table consists of several lines. The line with the
index number at the left hand margin lists the current function.
The lines above it list the functions that called this function,
and the lines below it list the functions this one called.

This line lists:

index A unique number given to each element of the table.
 Index numbers are sorted numerically.
 The index number is printed next to every function name so
 it is easier to look up where the function is in the table.

% time This is the percentage of the 'total' time that was spent
 in this function and its children. Note that due to
 different viewpoints, functions excluded by options, etc,
 these numbers will NOT add up to 100%.

c:\Users\91826\projects\C\Calculator>gprof a.exe gmon.out > main_prof.txt

Debug (C++) :

main.cpp

c:\Users\91826\projects\C\Calculator>g++ main.cpp -g

c:\Users\91826\projects\C\Calculator>gdb a.exe

GNU gdb (GDB) 11.1

Copyright (C) 2021 Free Software Foundation, Inc.

License GPLv3+: GNU GPL version 3 or later <<http://gnu.org/licenses/gpl.html>>

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

Type "show copying" and "show warranty" for details.

This GDB was configured as "x86_64-w64-mingw32".

Type "show configuration" for configuration details.

For bug reporting instructions, please see:

<<https://www.gnu.org/software/gdb/bugs/>>.

Find the GDB manual and other documentation resources online at:

<<http://www.gnu.org/software/gdb/documentation/>>.


```

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.exe...
(gdb) run
Starting program: c:\Users\91826\projects\C\Calculator\a.exe
[New Thread 9688.0x3974]
[New Thread 9688.0x3460]
[New Thread 9688.0x40b4]
Enter the operation you want to perform :
add
Enter the 2 numbers :
58 68
Sum : 126
Do you want to continue (y/n) : n
Calculator closed!!
[Thread 9688.0x40b4 exited with code 0]
[Thread 9688.0x3460 exited with code 0]
[Thread 9688.0x5b8 exited with code 0]
[Inferior 1 (process 9688) exited normally]
(gdb) break 30
No line 30 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) break 11
Breakpoint 1 at 0x7ff6835c1f3a: file main.cpp, line 11.
(gdb) run
Starting program: c:\Users\91826\projects\C\Calculator\a.exe
[New Thread 14864.0xa30]
[New Thread 14864.0x4df0]
[New Thread 14864.0x4128]

Thread 1 hit Breakpoint 1, main () at main.cpp:11
warning: Source file is more recent than executable.
11         interface();
(gdb) n
Enter the operation you want to perform :
sub
Enter the 2 numbers :
58 68
Difference : -1012          cout << "\nDo you want to continue (y/n) : ";
(gdb) n

Do you want to continue (y/n) : 13          cin >> cont;
(gdb) n
n
14         f = (cont == "y") ? true : false;
(gdb) n
9         while (f)
(gdb) n
17         cout << "Calculator closed!!\n";
(gdb) n
Calculator closed!!
18         return 0;
(gdb) step
19     }
(gdb) jump 2
Continuing at 0x7ff6835c1f02.

```

```

Thread 1 hit Breakpoint 1, main () at main.cpp:11
11      interface();
(gdb) break 15
Breakpoint 2 at 0x7ff6835c1f8a: file main.cpp, line 17.
(gdb) break
Note: breakpoint 1 also set at pc 0x7ff6835c1f3a.
Breakpoint 3 at 0x7ff6835c1f3a: file main.cpp, line 11.
(gdb) info break
Num      Type           Disp Enb Address            What
1        breakpoint      keep y  0x00007ff6835c1f3a in main() at main.cpp:11
        breakpoint already hit 2 times
2        breakpoint      keep y  0x00007ff6835c1f8a in main() at main.cpp:17
3        breakpoint      keep y  0x00007ff6835c1f3a main.cpp:11
(gdb) disable 2
(gdb) info break
Num      Type           Disp Enb Address            What
1        breakpoint      keep y  0x00007ff6835c1f3a in main() at main.cpp:11
        breakpoint already hit 2 times
2        breakpoint      keep n  0x00007ff6835c1f8a in main() at main.cpp:17
3        breakpoint      keep y  0x00007ff6835c1f3a main.cpp:11
(gdb) disable 3
(gdb) info break
Num      Type           Disp Enb Address            What
1        breakpoint      keep y  0x00007ff6835c1f3a in main() at main.cpp:11
        breakpoint already hit 2 times
2        breakpoint      keep n  0x00007ff6835c1f8a in main() at main.cpp:17
3        breakpoint      keep n  0x00007ff6835c1f3a main.cpp:11
(gdb) enable 3
(gdb) info break
Num      Type           Disp Enb Address            What
1        breakpoint      keep y  0x00007ff6835c1f3a in main() at main.cpp:11
        breakpoint already hit 2 times
2        breakpoint      keep n  0x00007ff6835c1f8a in main() at main.cpp:17
3        breakpoint      keep y  0x00007ff6835c1f3a main.cpp:11
(gdb) continue
Continuing.
Enter the operation you want to perform :
mul
Enter the 2 numbers :
68 54
Product : 3672
Do you want to continue (y/n) : y

Thread 1 hit Breakpoint 1, main () at main.cpp:11
11      interface();
(gdb) del 1
(gdb) info break
Num      Type           Disp Enb Address            What
2        breakpoint      keep n  0x00007ff6835c1f8a in main() at main.cpp:17
3        breakpoint      keep y  0x00007ff6835c1f3a main.cpp:11
        breakpoint already hit 1 time
(gdb) print f
$1 = true

```

```
(gdb) c
Continuing.
Enter the operation you want to perform :
fact
Enter 1 positive integer to find factorial :
7
5040
Do you want to continue (y/n) : n
Calculator closed!!
[Thread 14864.0x4df0 exited with code 0]
[Thread 14864.0xa30 exited with code 0]
[Thread 14864.0x4128 exited with code 0]
[Inferior 1 (process 14864) exited normally]
(gdb) quit
```

Code (C++) :

main.cpp

```
#include "interface.h"

// main function
int main()
{
    bool f = true;
    string cont = "y";
    // calling interface function to interact with the user
    while (f)
    {
        interface();
        cout << "\nDo you want to continue (y/n) : ";
        cin >> cont;
        f = (cont == "y") ? true : false;
    }

    cout << "Calculator closed!!\n";
    return 0;
}
```

interface.h

```
#include "operations.h"

void check_op(string op);

// output function to do operations and show on the terminal
void interface()
{
    // string object op to input the operation
    string op;
    cout << "Enter the operation you want to perform : \n";
    cin >> op;
```

```

        // calling function check
        check_op(op);
    }

    // function to check the operation entered by the user
    void check_op(string op)
    {
        // checking if operation is addition
        if (op == "add")
            add();
        // checking if operation is subtraction
        else if (op == "sub")
            sub();
        // checking if operation is multiply
        else if (op == "mul")
            mul();
        // checking if operation is division
        else if (op == "div")
            div();
        // checking if operation to find percentage
        else if (op == "percent")
            percent();
        // checking if operation is factorial
        else if (op == "fact")
        {
            int num1;
            long long unsigned int fac_res;
            cout << "Enter 1 positive integer to find factorial:\n";
            cin >> num1;
            fac_res = fact(num1);
            cout << fac_res;
        }
        // checking if operation is nPr
        else if (op == "nPr")
        {
            int num1, num2;
            long long unsigned int nPr_res;
            cout << "Enter 2 positive numbers:\n";
            cin >> num1 >> num2;
            nPr_res = nPr(num1, num2);
            cout << nPr_res;
        }
        // checking if operation is nCr
        else if (op == "nCr")
            nCr();
        // checking if operation is finding square
    }

```

```

    else if (op == "sqr")
        sqr();
    // checking if operation is to find cube
    else if (op == "cube")
        cube();
    // checking if operation is to find  $n^k$ 
    else if (op == "pow")
        power();
    // checking if operation is sine
    else if (op == "sin")
        sine();
    // checking if operation is cosine
    else if (op == "cos")
        cosine();
    // checking if operation is tangent
    else if (op == "tan")
        tangent();
    else
        cout << "SORRY!! I can't do this.\n";
}

```

operations.h

```

#include <iostream>
#include <math.h>
#include <stdlib.h>
#include <stdio.h>

using namespace std;

// function to add 2 numbers
void add()
{
    double num1, num2;
    cout << "Enter the 2 numbers:\n";
    cin >> num1 >> num2;

    cout << "Sum:" << num1 + num2;
}

// function to subtract 2 numbers
void sub()
{
    double num1, num2;
    cout << "Enter the 2 numbers:\n";
    cin >> num1 >> num2;

    cout << "Difference:" << num1 - num2;
}

```

```
// function to multiply 2 numbers
void mul()
{
    double num1, num2;
    cout << "Enter the 2 numbers:\n";
    cin >> num1 >> num2;

    cout << "Product: " << num1 * num2;
}

// function to divide 2 numbers and give the output in decimal
void div()
{
    double num1, num2;
    cout << "Enter the 2 numbers:\n";
    cin >> num1 >> num2;

    cout << "Division: " << num1 / num2;
}

// function to find percentage of a number
void percent()
{
    double num1, num2;
    cout << "Enter 2 number:\n";
    cin >> num1 >> num2;

    cout << num1 * num2 / 100;
}

// function to find factorial
long long unsigned int fact(int num1)
{
    long long unsigned int fac_res = 1;

    while (num1 > 1)
    {
        fac_res *= num1;
        num1 -= 1;
    }
    return fac_res;
}

// Function to find the number of permutations of n
// objects taking r at a time, that is nPr
long long unsigned int nPr(int num1, int num2)
{

```

```
    int temp;
    long long unsigned int nPr_res;

    if (num1 < num2)
    {
        temp = num1;
        num1 = num2;
        num2 = temp;
    }

    nPr_res = fact(num1) / fact(num1 - num2);
    return nPr_res;
}

// Function to find the number of combination of how n
// objects can be selected taking r at a time, that is nCr
void nCr()
{
    int num1, num2, temp;
    cout << "Enter 2 positive numbers:\n";
    cin >> num1 >> num2;

    if (num1 < num2)
    {
        temp = num1;
        num1 = num2;
        num2 = temp;
    }

    if (num1 == num2)
        cout << "nCr: " << 1;
    else
        cout << "nCr: " << nPr(num1, num2) / fact(num2);
}

// Function to do square of a number
void sqr()
{
    double num;
    cout << "Enter a number:\n";
    cin >> num;

    cout << num * num;
}

// Function to do cube of a number
void cube()
{

```



```
    double num;
    cout << "Enter a number: \n";
    cin >> num;

    cout << num * num * num;
}

// Function to get answer of number to the power n
void power()
{
    double num1, num2;
    cout << "Enter the number and its power: \n";
    cin >> num1 >> num2;

    cout << pow(num1, num2);
}

// Function to find sine of a given number
void sine()
{
    double num;
    cout << "Enter the number: \n";
    cin >> num;

    cout << sin(num);
}

// Function to find cosine of a given number
void cosine()
{
    double num;
    cout << "Enter the number: \n";
    cin >> num;

    cout << cos(num);
}

// Function to find tangent of a given number
void tangent()
{
    double num;
    cout << "Enter the number: \n";
    cin >> num;

    cout << tan(num);
}
```

Ouput (C++) :

```
c:\Users\91826\projects\C\Calculator>g++ main.cpp
```

```
c:\Users\91826\projects\C\Calculator>a.exe
```

```
Enter the operation you want to perform :
```

```
add
```

```
Enter the 2 numbers :
```

```
5 46
```

```
Sum : 51
```

```
Do you want to continue (y/n) : y
```

```
Enter the operation you want to perform :
```

```
sub
```

```
Enter the 2 numbers :
```

```
57 69
```

```
Difference : -12
```

```
Do you want to continue (y/n) : y
```

```
Enter the operation you want to perform :
```

```
mul
```

```
Enter the 2 numbers :
```

```
58 47
```

```
Product : 2726
```

```
Do you want to continue (y/n) : y
```

```
Enter the operation you want to perform :
```

```
div
```

```
Enter the 2 numbers :
```

```
5 4
```

```
Division : 1.25
```

```
Do you want to continue (y/n) : y
```

```
Enter the operation you want to perform :
```

```
percent
```

```
Enter 2 number :
```

```
50 2
```

```
1
```

```
Do you want to continue (y/n) : y
```

```
Enter the operation you want to perform :
```

```
fact
```

```
Enter 1 positive integer to find factorial :
```

```
6
```

```
720
```

```
Do you want to continue (y/n) : y
```

```
Enter the operation you want to perform :
```

```
nPr
```

```
Enter 2 positive numbers :
```

```
6 2
```

```
30
```

```
Do you want to continue (y/n) : y
```

```
Enter the operation you want to perform :
```

```
nCr
```

```
Enter 2 positive numbers :
```

```
6 2
```

```
nCr : 15
```

```
Do you want to continue (y/n) : y
```

```
Enter the operation you want to perform :
```

```
sqr
```

```
Enter a number :
```

```
5
```

```
25
```

```
Do you want to continue (y/n) : y
```

```
Enter the operation you want to perform :
cube
Enter a number :
6
216
Do you want to continue (y/n) : y
Enter the operation you want to perform :
pow
Enter the number and its power :
8 -.5
0.353553
Do you want to continue (y/n) : y
Enter the operation you want to perform :
sin
Enter the number :
3.14
0.00159265
Do you want to continue (y/n) : y
Enter the operation you want to perform :
cos
Enter the number :
3.1415
-1
Do you want to continue (y/n) : y
Enter the operation you want to perform :
tan
Enter the number :
.7854
1
Do you want to continue (y/n) : n
Calculator closed!!

Enter the operation you want to perform :
cosine
SORRY!! I can't do this.

Do you want to continue (y/n) : n
Calculator closed!!
```