

2303A52416

A.akshaya

Problem 1:

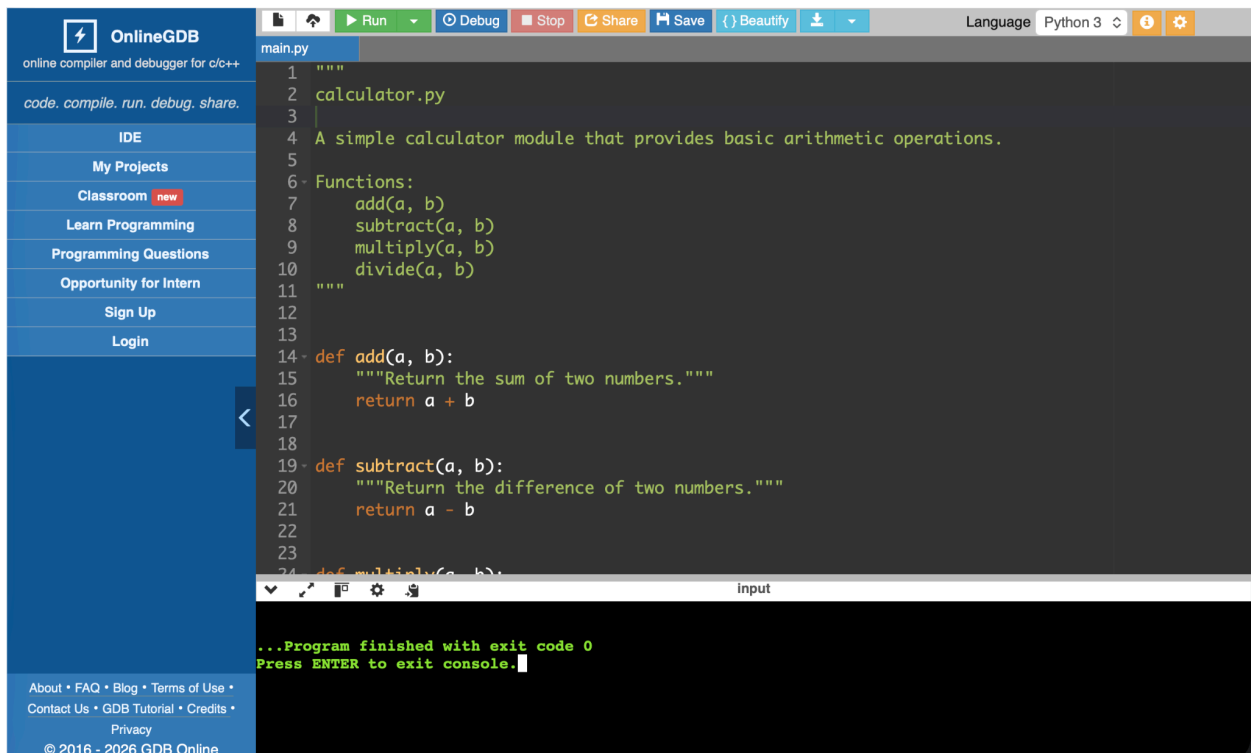
Consider the following Python function:

```
def find_max(numbers):  
    return max(numbers)
```

Task:

- Write documentation for the function in all three formats:
 - (a) Docstring
 - (b) Inline comments
 - (c) Google-style documentation
- Critically compare the three approaches. Discuss the advantages, disadvantages, and suitable use cases of each style.
- Recommend which documentation style is most effective for a mathematical utilities library and justify your Answer.

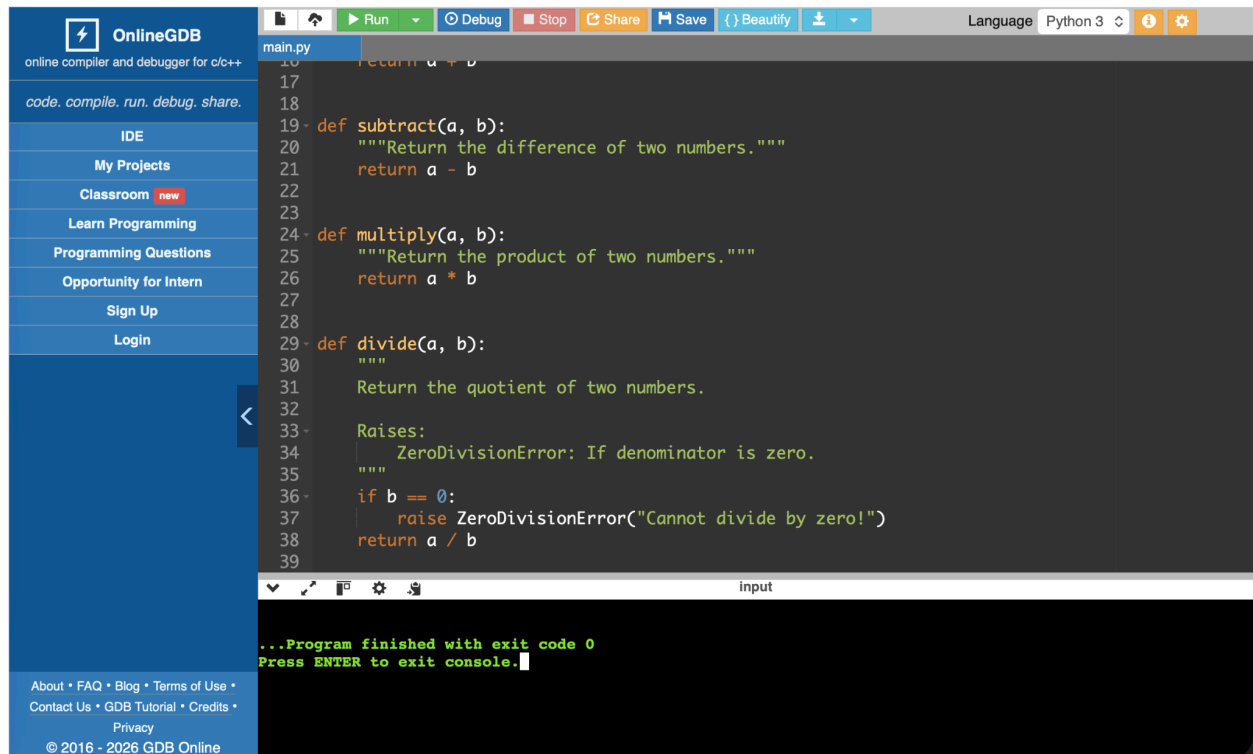
Code



The screenshot shows the OnlineGDB IDE interface. The left sidebar contains navigation links: IDE, My Projects, Classroom (new), Learn Programming, Programming Questions, Opportunity for Intern, Sign Up, and Login. The main editor area displays a Python file named 'main.py' with the following code:

```
1 """  
2 calculator.py  
3  
4 A simple calculator module that provides basic arithmetic operations.  
5  
6 Functions:  
7     add(a, b)  
8     subtract(a, b)  
9     multiply(a, b)  
10    divide(a, b)  
11 """  
12  
13  
14 def add(a, b):  
15     """Return the sum of two numbers."""  
16     return a + b  
17  
18  
19 def subtract(a, b):  
20     """Return the difference of two numbers."""  
21     return a - b  
22  
23  
24 def multiply(a, b):  
25     """Return the product of two numbers."""  
26     return a * b  
27  
28 def divide(a, b):  
29     """Return the quotient of two numbers."""  
30     return a / b  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

The bottom console shows the output: "...Program finished with exit code 0" and "Press ENTER to exit console."



The screenshot shows the OnlineGDB web interface. On the left is a blue sidebar with navigation links: IDE, My Projects, Classroom (marked 'new'), Learn Programming, Programming Questions, Opportunity for Intern, Sign Up, and Login. At the bottom of the sidebar are links for About, FAQ, Blog, Terms of Use, Contact Us, GDB Tutorial, Credits, and Privacy, along with a copyright notice for 2016-2026 GDB Online. The main area is a dark-themed code editor with a file named 'main.py'. It contains four Python functions: 'add(a, b)', 'subtract(a, b)', 'multiply(a, b)', and 'divide(a, b)'. The 'divide' function includes a docstring with a 'Raises:' section for 'ZeroDivisionError' and a conditional check 'if b == 0:' to raise the error. The top of the editor has a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to 'Python 3'. At the bottom, an input field and a console output area are visible, showing the message '...Program finished with exit code 0'.

```
16 def add(a, b):
17     return a + b
18
19 def subtract(a, b):
20     """Return the difference of two numbers."""
21     return a - b
22
23
24 def multiply(a, b):
25     """Return the product of two numbers."""
26     return a * b
27
28
29 def divide(a, b):
30     """
31     Return the quotient of two numbers.
32
33     Raises:
34         ZeroDivisionError: If denominator is zero.
35     """
36     if b == 0:
37         raise ZeroDivisionError("Cannot divide by zero!")
38     return a / b
39
```

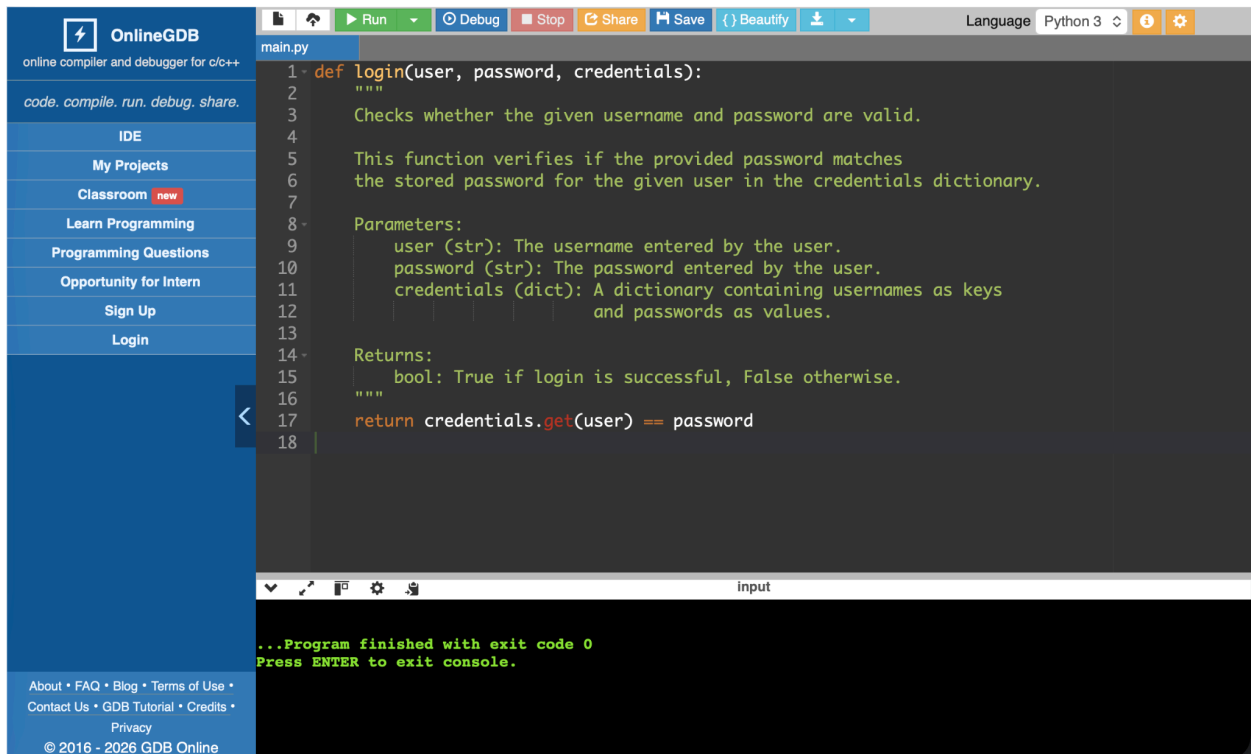
...Program finished with exit code 0
Press ENTER to exit console.

Problem 2: Consider the following Python function:

```
def login(user, password, credentials):  
    return credentials.get(user) == password
```

Task:

1. Write documentation in all three formats.
2. Critically compare the approaches.
3. Recommend which style would be most helpful for new developers onboarding a project, and justify your choice.



The screenshot shows the OnlineGDB web interface. On the left is a blue sidebar with navigation links: IDE, My Projects, Classroom (marked 'new'), Learn Programming, Programming Questions, Opportunity for Intern, Sign Up, and Login. At the bottom of the sidebar are links for About, FAQ, Blog, Terms of Use, Contact Us, GDB Tutorial, Credits, and Privacy, along with a copyright notice for 2016-2026 GDB Online. The main editor area has a top toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to Python 3. The code in the editor is a Python function named 'login' that takes 'user', 'password', and 'credentials' as arguments. It includes a docstring with a description, parameters, and return value. The function uses 'credentials.get(user) == password' to check the login. Below the editor is an 'Input' field and a console output showing '...Program finished with exit code 0' and 'Press ENTER to exit console.'

```
1 def login(user, password, credentials):
2     """
3     Checks whether the given username and password are valid.
4
5     This function verifies if the provided password matches
6     the stored password for the given user in the credentials dictionary.
7
8     Parameters:
9         user (str): The username entered by the user.
10        password (str): The password entered by the user.
11        credentials (dict): A dictionary containing usernames as keys
12                           and passwords as values.
13
14    Returns:
15        bool: True if login is successful, False otherwise.
16    """
17    return credentials.get(user) == password
18
```

...Program finished with exit code 0
Press ENTER to exit console.

Problem 3: Calculator (Automatic Documentation Generation)

Task: Design a Python module named `calculator.py` and demonstrate automatic documentation generation.

Instructions:

1. Create a Python module `calculator.py` that includes the following functions, each written with appropriate docstrings:
 - o `add(a, b)` – returns the sum of two numbers

- o `subtract(a, b)` – returns the difference of two numbers

- o `multiply(a, b)` – returns the product of two numbers

- o `divide(a, b)` – returns the quotient of two numbers


2. Display the module documentation in the terminal using

Python's documentation tools.

3. Generate and export the module documentation in HTML

format using the `pydoc` utility, and open the generated HTML

file in a web browser to verify the output.

**OnlineGDB**

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions










Opportunity for Intern

Sign Up

Login

About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy

© 2016 - 2026 GDB Online


 Run  Debug  Stop  Share  Save  Beautify 

main.py

```
1  """
2  calculator.py
3
4  A simple Calculator Module.
5
6  This module provides basic arithmetic operations such as:
7
8      - Addition
9      - Subtraction
10     - Multiplication
11     - Division
12
13 Each function includes proper docstrings so that documentation
14 can be generated automatically using Python tools like pydoc.
15 """
16
17
18 def add(a, b):
19     """
20     Add two numbers.
21
22     Args:
23         a (int/float): First number
24         b (int/float): Second number
25
26     Returns:
27         int/float: The sum of a and b
28     """
29     return a + b
30
31
32 def subtract(a, b):
33     """
34     Subtract b from a.
35
36     Args:
37         a (int/float): First number
38         b (int/float): Second number
39
40     Returns:
41         int/float: The difference of a and b
42     """
43     return a - b
44
45
46 def multiply(a, b):
47     """
48     Multiply a and b.
49
50     Args:
51         a (int/float): First number
52         b (int/float): Second number
53
54     Returns:
55         int/float: The product of a and b
56     """
57     return a * b
58
59
60 def divide(a, b):
61     """
62     Divide a by b.
63
64     Args:
65         a (int/float): First number
66         b (int/float): Second number
67
68     Returns:
69         int/float: The quotient of a and b
70     """
71     return a / b
72
73
74 def main():
75     """
76     Main function to run the calculator.
77     """
78     print("Calculator Module")
79     print("Operations: add, subtract, multiply, divide")
80     print("Enter numbers and operation to perform.")
81     a = float(input("Enter first number: "))
82     b = float(input("Enter second number: "))
83     op = input("Enter operation (add, subtract, multiply, divide): ")
84
85     if op == "add":
86         result = add(a, b)
87     elif op == "subtract":
88         result = subtract(a, b)
89     elif op == "multiply":
90         result = multiply(a, b)
91     elif op == "divide":
92         result = divide(a, b)
93     else:
94         print("Invalid operation")
95         return
96
97     print(f"Result: {result}")
98
99 if __name__ == "__main__":
100    main()
```

input

...Program finished with exit code 0
Press ENTER to exit console.

**OnlineGDB**

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions

Opportunity for Intern

Sign Up

Login

About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy


© 2016 - 2026 GDB Online

main.py

```
24         b (int/float): Second number
25
26     Returns:
27         int/float: Sum of a and b
28     """
29     return a + b
30
31
32 def subtract(a, b):
33     """
34     Subtract two numbers.
35
36     Args:
37         a (int/float): First number
38         b (int/float): Second number
39
40     Returns:
41         int/float: Difference of a and b
42     """
43     return a - b
44
45
46 def multiply(a, b):
47     """
```

input

...Program finished with exit code 0
Press ENTER to exit console.

**OnlineGDB**

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions

Opportunity for Intern

Sign Up

Login

About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy

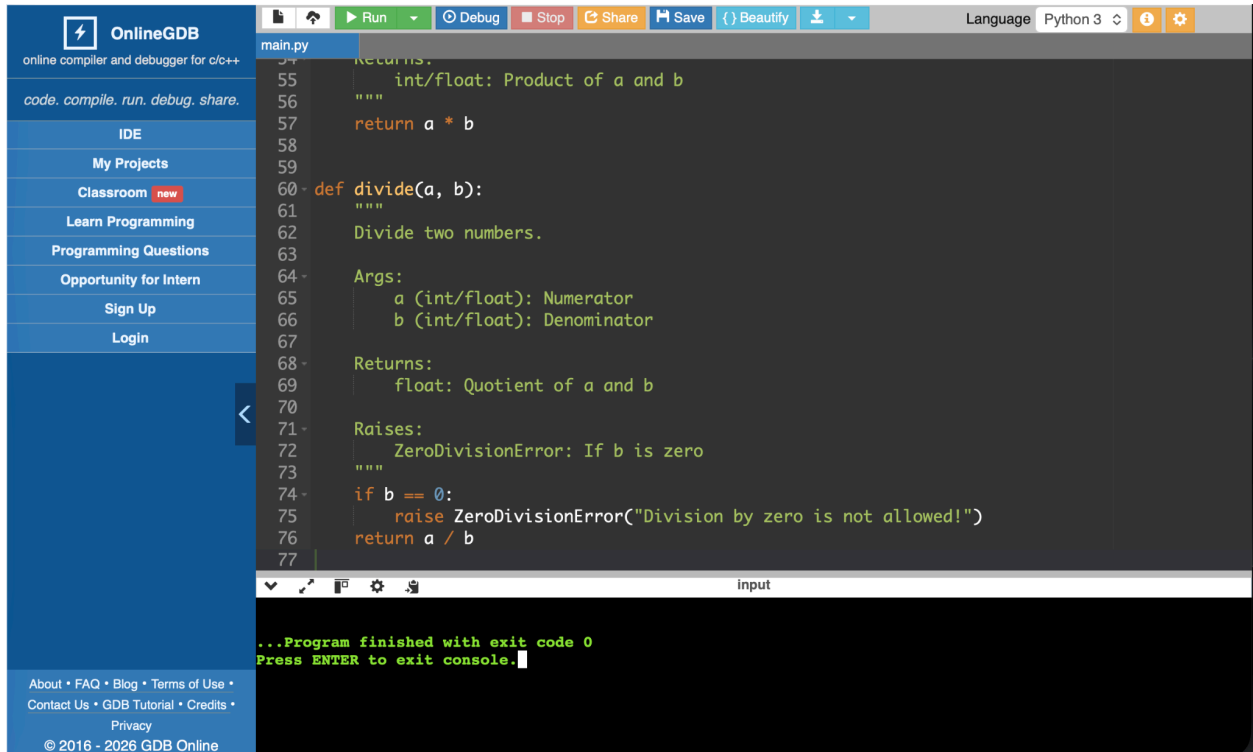
© 2016 - 2026 GDB Online

main.py

```
47     """
48     Multiply two numbers.
49
50     Args:
51         a (int/float): First number
52         b (int/float): Second number
53
54     Returns:
55         int/float: Product of a and b
56     """
57     return a * b
58
59
60 def divide(a, b):
61     """
62     Divide two numbers.
63
64     Args:
65         a (int/float): Numerator
66         b (int/float): Denominator
67
68     Returns:
69         float: Quotient of a and b
70     """
```

input

...Program finished with exit code 0
Press ENTER to exit console.



The screenshot shows the OnlineGDB web interface. On the left is a sidebar with navigation links: IDE, My Projects, Classroom (marked 'new'), Learn Programming, Programming Questions, Opportunity for Intern, Sign Up, and Login. At the bottom of the sidebar are links for About, FAQ, Blog, Terms of Use, Contact Us, GDB Tutorial, Credits, and Privacy, along with a copyright notice for 2016-2026 GDB Online. The main editor area displays a file named 'main.py' with the following Python code:


```
54 returns:
55     int/float: Product of a and b
56     """
57     return a * b
58
59
60 def divide(a, b):
61     """
62     Divide two numbers.
63
64     Args:
65         a (int/float): Numerator
66         b (int/float): Denominator
67
68     Returns:
69         float: Quotient of a and b
70
71     Raises:
72         ZeroDivisionError: If b is zero
73     """
74     if b == 0:
75         raise ZeroDivisionError("Division by zero is not allowed!")
76     return a / b
77
```

Below the code editor is an 'input' field and a terminal window showing the output: "...Program finished with exit code 0" and "Press ENTER to exit console."

Problem 4: Conversion Utilities Module

Task:

1. Write a module named `conversion.py` with functions:
 - o `decimal_to_binary(n)`
 - o `binary_to_decimal(b)`
 - o `decimal_to_hexadecimal(n)`
2. Use Copilot for auto-generating docstrings.
3. Generate documentation in the terminal.
4. Export the documentation in HTML format and open it in a browser.

**OnlineGDB**
online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming








Programming Questions

Opportunity for Intern

Sign Up

Login

About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy
© 2016 - 2026 GDB Online




main.py

```
1 """
2 conversion.py
3
4 A Conversion Utilities Module.
5
6 This module provides functions to convert numbers between
7 different number systems:
8
9     - Decimal to Binary
10    - Binary to Decimal
11    - Decimal to Hexadecimal
12
13 Docstrings are included so that documentation can be
14 generated automatically using pydoc.
15 """
16
17
18 def decimal_to_binary(n):
19     """
20     Convert a decimal integer to its binary representation.
21
22     Args:
23         n (int): A decimal number
24
25     Returns:
26         str: Binary string equivalent of the decimal number
27
28     Example:
29         decimal_to_binary(10) -> '1010'
30
31     """
32     return bin(n)[2:]
33
34 def binary_to_decimal(b):
35     """
36     Convert a binary string to its decimal representation.
37
38     Args:
39         b (str): A binary number as a string
40
41     Returns:
42         int: Decimal equivalent of the binary number
43
44     Example:
45         binary_to_decimal("1010") -> 10
46
47     """
48     return int(b, 2)
```

input

...Program finished with exit code 0
Press ENTER to exit console.

**OnlineGDB**
online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming








Programming Questions

Opportunity for Intern

Sign Up

Login

About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy
© 2016 - 2026 GDB Online



main.py

```
25 Returns:
26     str: Binary string equivalent of the decimal number
27
28 Example:
29     decimal_to_binary(10) -> '1010'
30 """
31 return bin(n)[2:]
32
33
34 def binary_to_decimal(b):
35     """
36     Convert a binary string to its decimal representation.
37
38     Args:
39         b (str): A binary number as a string
40
41     Returns:
42         int: Decimal equivalent of the binary number
43
44     Example:
45         binary_to_decimal("1010") -> 10
46
47     """
48     return int(b, 2)
```

input

...Program finished with exit code 0
Press ENTER to exit console.

The screenshot shows the OnlineGDB IDE interface. On the left is a sidebar with navigation links: IDE, My Projects, Classroom (marked 'new'), Learn Programming, Programming Questions, Opportunity for Intern, Sign Up, and Login. At the bottom of the sidebar are links for About, FAQ, Blog, Terms of Use, Contact Us, GDB Tutorial, Credits, and Privacy, along with the copyright notice © 2016 - 2026 GDB Online. The main editor area displays a file named 'main.py' with the following Python code:


```
41 returns:
42     int: Decimal equivalent of the binary number
43
44 Example:
45     binary_to_decimal("1010") -> 10
46     """
47     return int(b, 2)
48
49
50 def decimal_to_hexadecimal(n):
51     """
52     Convert a decimal integer to its hexadecimal representation.
53
54     Args:
55         n (int): A decimal number
56
57     Returns:
58         str: Hexadecimal string equivalent of the decimal number
59
60     Example:
61         decimal_to_hexadecimal(255) -> 'FF'
62     """
63     return hex(n)[2:].upper()
64
```

Below the code editor is an input field and a terminal window. The terminal shows the output: "...Program finished with exit code 0" and "Press ENTER to exit console."

Problem 5 – Course Management Module

Task:

1. Create a module `course.py` with functions:
 - o `add_course(course_id, name, credits)`
 - o `remove_course(course_id)`
 - o `get_course(course_id)`
2. Add docstrings with Copilot.
3. Generate documentation in the terminal.
4. Export the documentation in HTML format and open it in a browser.

**OnlineGDB**

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming








Programming Questions

Opportunity for Intern

Sign Up

Login

About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy
© 2016 - 2026 GDB Online




main.py

```
1  """
2  course.py
3
4  Course Management Module.
5
6  This module provides basic functions to manage courses
7  in a simple course database.
8
9  Functions included:
10
11     - add_course(course_id, name, credits)
12     - remove_course(course_id)
13     - get_course(course_id)
14
15  Docstrings are added so that documentation can be
16  automatically generated using Python tools like pydoc.
17  """
18
19  # Dictionary to store course details
20  courses = {}
21
22
23  def add_course(course_id, name, credits):
24      """
```

input

...Program finished with exit code 0
Press ENTER to exit console.

**OnlineGDB**

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming








Programming Questions

Opportunity for Intern

Sign Up

Login

About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy
© 2016 - 2026 GDB Online




main.py

```
22
23  def add_course(course_id, name, credits):
24      """
25      Add a new course into the course database.
26
27      Args:
28          course_id (str): Unique identifier for the course
29          name (str): Name of the course
30          credits (int): Number of credits assigned to the course
31
32      Returns:
33          str: Confirmation message after adding the course
34      """
35      courses[course_id] = {
36          "name": name,
37          "credits": credits
38      }
39      return f"Course {course_id} added successfully!"
40
41
42  def remove_course(course_id):
43      """
44      Remove an existing course from the course database.
45
```

input

...Program finished with exit code 0
Press ENTER to exit console.

**OnlineGDB**

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions










Opportunity for Intern

Sign Up


Login



About • FAQ • Blog • Terms of Use •
Contact Us • GDB Tutorial • Credits •
Privacy

© 2016 - 2026 GDB Online

 Run Debug Stop Share Save Beautify

main.py

Language Python 3 



```
46 - Args:
47     course_id (str): Unique identifier of the course to remove
48
49 - Returns:
50     str: Confirmation message if removed, otherwise error message
51     """
52 - if course_id in courses:
53     del courses[course_id]
54     return f"Course {course_id} removed successfully!"
55     return "Course not found!"
56
57
58 - def get_course(course_id):
59     """
60     Retrieve details of a specific course.
61
62     Args:
63         course_id (str): Unique identifier of the course
64
65     Returns:
66         dict: Course information if found
67         str: Error message if course does not exist
68     """
69
70     # TODO: Implement the logic to retrieve course details
71     # based on the course_id.
72
73     # For now, returning a placeholder message.
74     return "Course not found!"
75
76
77 # Example usage:
78 if __name__ == '__main__':
79     # Get course_id from user input
80     course_id = input("Enter course ID: ")
81
82     # Call the get_course function
83     result = get_course(course_id)
84
85     # Print the result
86     print(result)
```

...Program finished with exit code 0
Press ENTER to exit console.