

INDEX

S.NO	PROGRAMS	SIGN.
1.	WAP to implement 'Inline function'.	
2.	WAP to implement call by reference and return by reference using class. [Hint. Assume necessary functions].	
3.	WAP to implement friend function by taking some real life example.	
4.	WAP to implement 'Function Overloading'.	
5.	WAP to implement Parameterized Constructor, Copy Constructor and Destructor.	
6.	WAP to show the usage of constructor in base and derived classes, in multiple inheritance.	
7.	WAP to show the implementation of 'containership'.	
8.	WAP to show swapping using template function (Generic).	
9.	WAP to implement 'Exception Handling'.	
10.	WAP to read and write values through object using file handling.	
11.	Create a class employee which have name, age and address of employee, include functions getdata() and showdata(), getdata() takes the input from the user, showdata() display the data in following format: Name: Age: Address:	
12.	WAP to add and subtract two complex numbers using classes.	
13.	Write program to overload Binary + to add two similar types of objects. (Both with and without using friend functions).	
14.	WAP to implement += and = operator.	
15.	WAP to convert meter to centimeter and vice versa, using data conversions and operator overloading.	
16.	WAP to count digits, alphabets and spaces, stored in a text file, using streams.	

PROGRAMS

1. WAP to implement 'Inline function'.

CODE: -

```
#include <iostream>
using namespace std;

inline void displayNum(int num) {
    cout << num << endl;
}

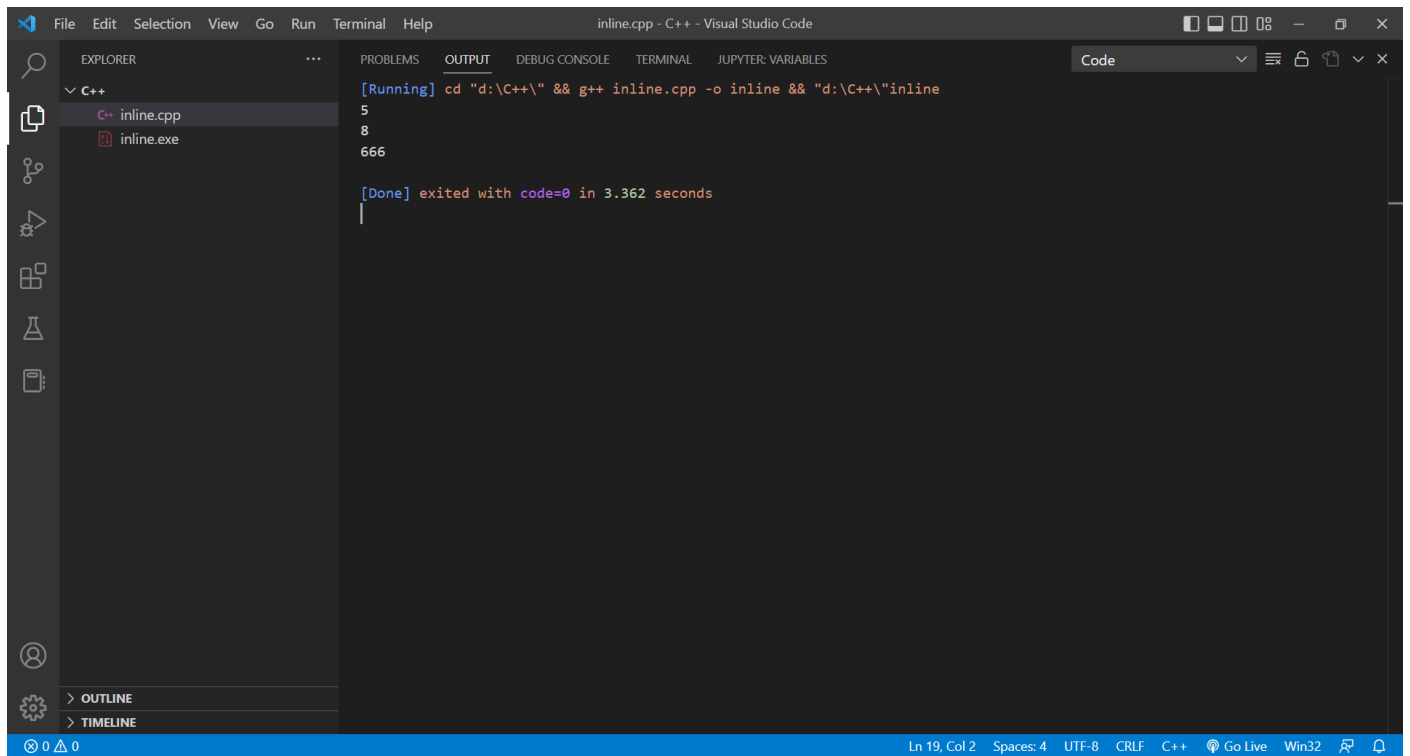
int main() {
    // first function call
    displayNum(5);

    // second function call
    displayNum(8);

    // third function call
    displayNum(666);

    return 0;
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'OUTPUT' panel active. The Explorer panel on the left shows a C++ project with files 'inline.cpp' and 'inline.exe'. The OUTPUT panel displays the execution output of the program, which prints the numbers 5, 8, and 666. The status bar at the bottom indicates the current line and column (Ln 19, Col 2) and the file encoding (UTF-8, CRLF).

```
File Edit Selection View Go Run Terminal Help inline.cpp - C++ - Visual Studio Code
```

EXPLORER

- ▼ C++
 - inline.cpp
 - inline.exe

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER: VARIABLES

Code

```
[Running] cd "d:\C++\" && g++ inline.cpp -o inline && "d:\C++\\"inline
5
8
666

[Done] exited with code=0 in 3.362 seconds
```

Ln 19, Col 2 Spaces: 4 UTF-8 CRLF C++ Go Live Win32

2. WAP to implement call by reference and return by reference using class. [Hint. Assume necessary functions].

CODE: -

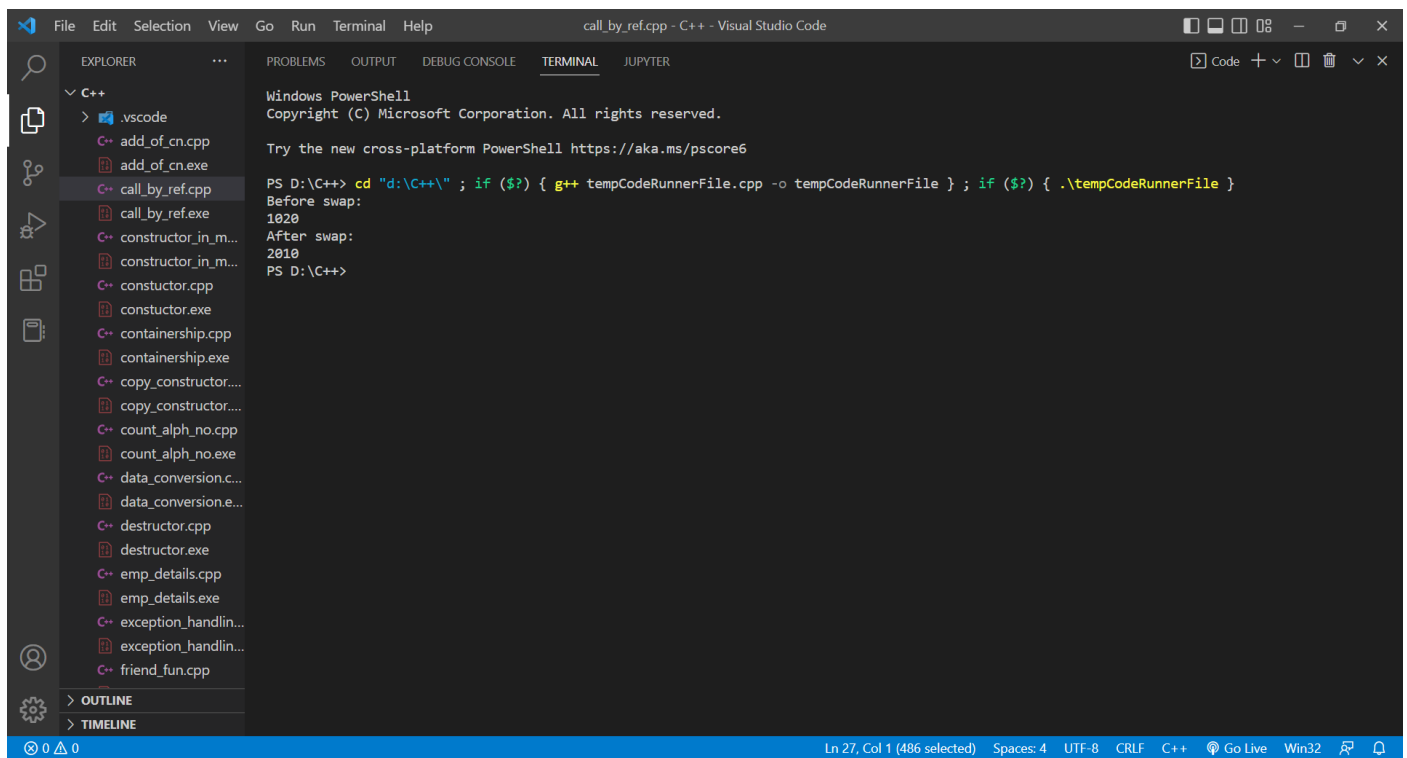
i.

```
#include <iostream>
using namespace std;
class A
{

    public:
    void swapNums(int &x, int &y)
    {
        int z = x;
        x = y;
        y = z;
    }

};
int main()
{
    int firstNum = 10;
    int secondNum = 20;
    A a1;
    cout << "Before swap: " << "\n";
    cout << firstNum << secondNum << "\n";
    a1.swapNums(firstNum, secondNum);
    cout << "After swap: " << "\n";
    cout << firstNum << secondNum << "\n";
    return 0;
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab active. The terminal displays the output of a C++ program executed in Windows PowerShell. The Explorer sidebar on the left lists various C++ files, with 'call_by_ref.cpp' selected. The terminal output shows the program's execution, including a swap operation and the resulting values.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Before swap:
1020
After swap:
2010
PS D:\C++>
```

The status bar at the bottom indicates the current line and column (Ln 27, Col 1), the number of selected characters (486), and the file encoding (UTF-8).

ii.

// C++ program to illustrate return by reference

```
#include <iostream>
```

```
using namespace std;
```

```
int a;
```

```
class A
```

```
{
```

```
    public:
```

```
    int& num()
```

```
    {
```

```
        return a;
```

```
    }
```

```
};
```

```
int main()
```

```
{
```

```
    A a1;
```

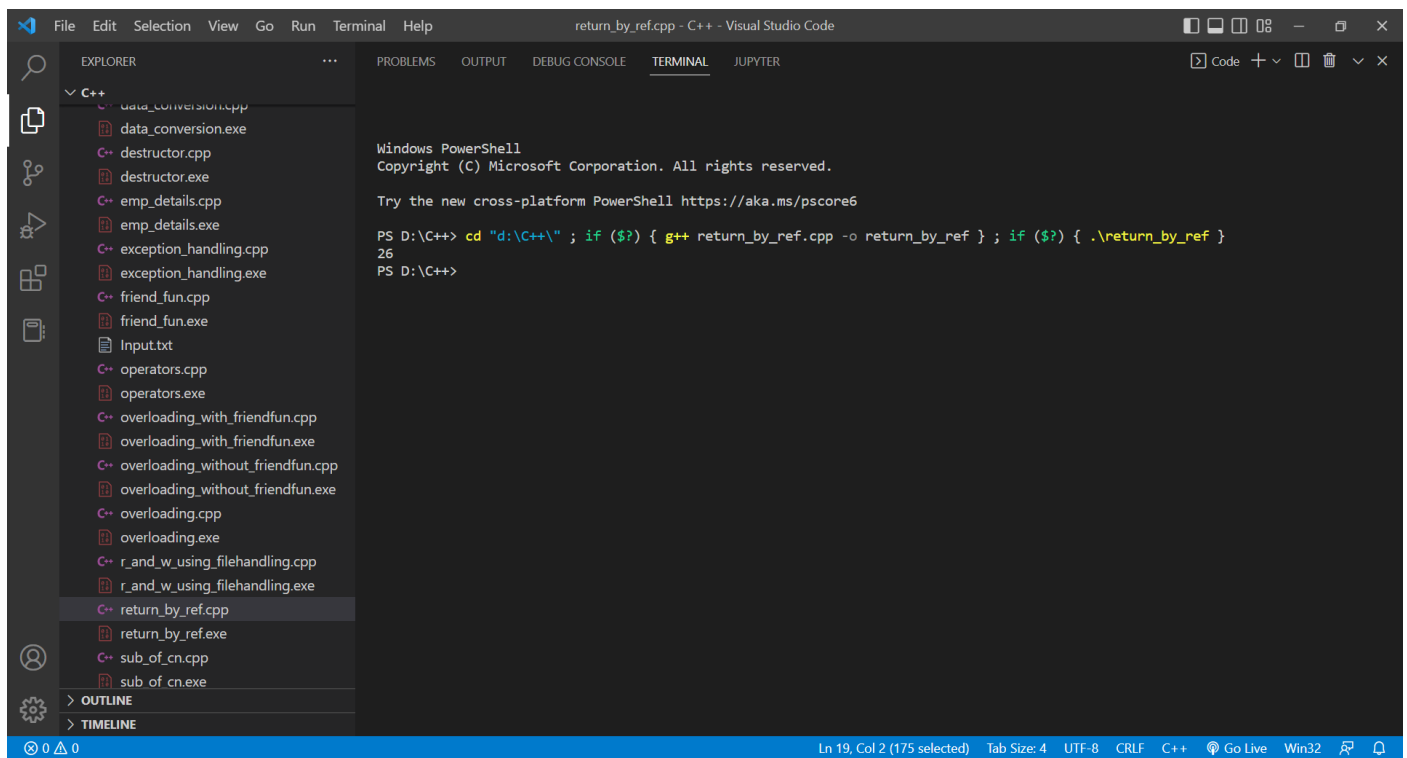
```
    a1.num()=26;
```

```
    cout<<a;
```

```
    return 0;
```

```
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left and the Terminal window on the right. The Explorer sidebar displays a list of files and folders for a C++ project, including source files (.cpp) and executables (.exe). The file 'return_by_ref.cpp' is selected. The Terminal window shows the output of a PowerShell command executed in the directory 'D:\C++'. The command is a batch script that compiles 'return_by_ref.cpp' using g++ and then runs the resulting executable. The output shows the Windows PowerShell prompt and the command execution path.

```
File Edit Selection View Go Run Terminal Help return_by_ref.cpp - C++ - Visual Studio Code
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
C++
  data_conversion.cpp
  data_conversion.exe
  destructor.cpp
  destructor.exe
  emp_details.cpp
  emp_details.exe
  exception_handling.cpp
  exception_handling.exe
  friend_fun.cpp
  friend_fun.exe
  Input.txt
  operators.cpp
  operators.exe
  overloading_with_friendfun.cpp
  overloading_with_friendfun.exe
  overloading_without_friendfun.cpp
  overloading_without_friendfun.exe
  overloading.cpp
  overloading.exe
  r_and_w_using_filehandling.cpp
  r_and_w_using_filehandling.exe
  return_by_ref.cpp
  return_by_ref.exe
  sub_of_cn.cpp
  sub_of_cn.exe
  > OUTLINE
  > TIMELINE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ return_by_ref.cpp -o return_by_ref } ; if ($?) { .\return_by_ref }
26
PS D:\C++>
```

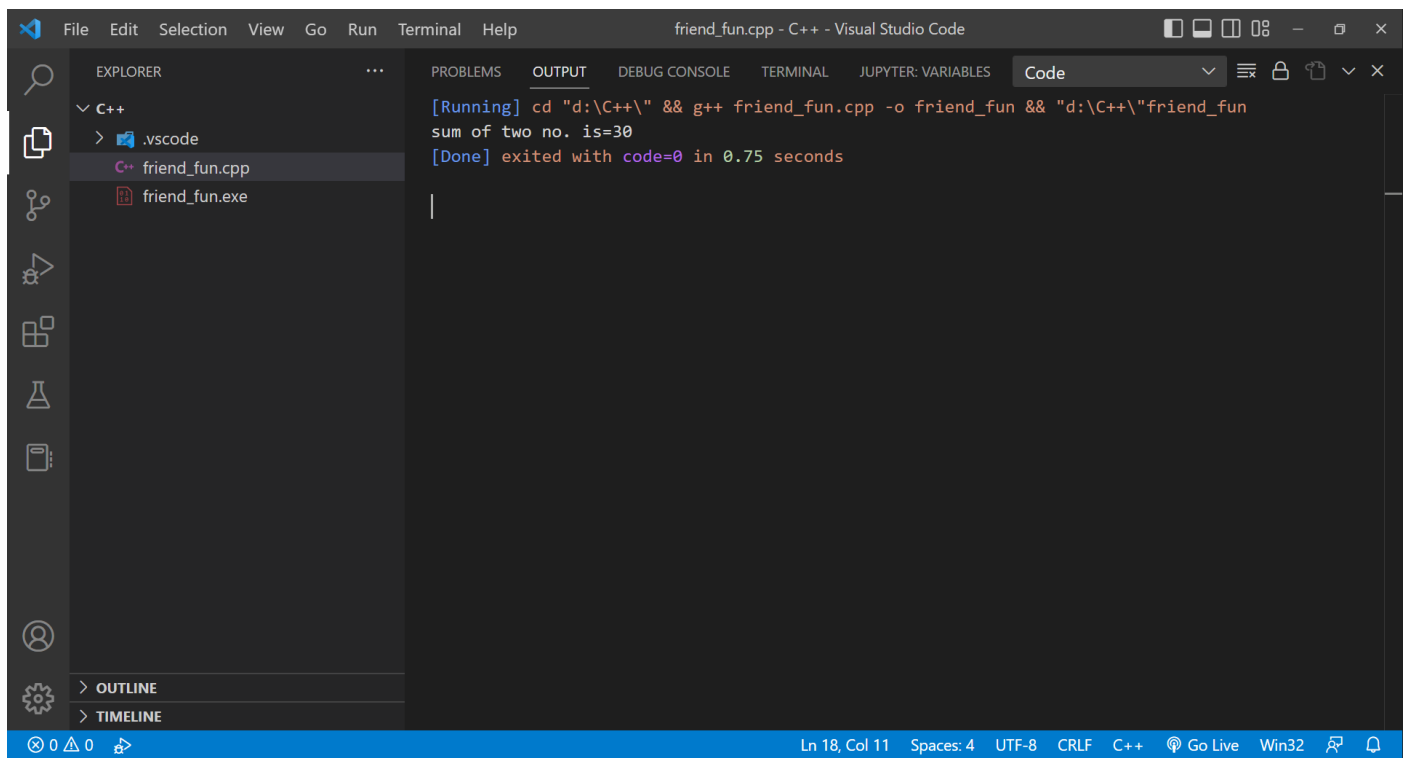
Ln 19, Col 2 (175 selected) Tab Size: 4 UTF-8 CRLF C++ Go Live Win32

3. WAP to implement friend function by taking some real-life example.

CODE: -

```
#include <iostream>
using namespace std;
class B;
class A
{
public:
    int x;
    friend void sum(A, B);
};
class B
{
public:
    int y;
    friend void sum(A, B);
};
void sum(A a1, B b1)
{
    int c;
    c = a1.x + b1.y;
    cout << "sum of two no. is=" << c;
}
int main()
{
    A a1;
    B b1;
    a1.x = 10;
    b1.y = 20;
    sum(a1, b1);
    return 0;
}
```


OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'OUTPUT' panel active. The Explorer panel on the left shows a project named 'C++' containing files '.vscode', 'friend_fun.cpp', and 'friend_fun.exe'. The OUTPUT panel displays the following text:

```
[Running] cd "d:\C++\" && g++ friend_fun.cpp -o friend_fun && "d:\C++\"friend_fun
sum of two no. is=30
[Done] exited with code=0 in 0.75 seconds
```

The status bar at the bottom indicates the current line and column as 'Ln 18, Col 11', with other details like 'Spaces: 4', 'UTF-8', 'CRLF', and 'C++'.

4. WAP to implement 'Function Overloading'.

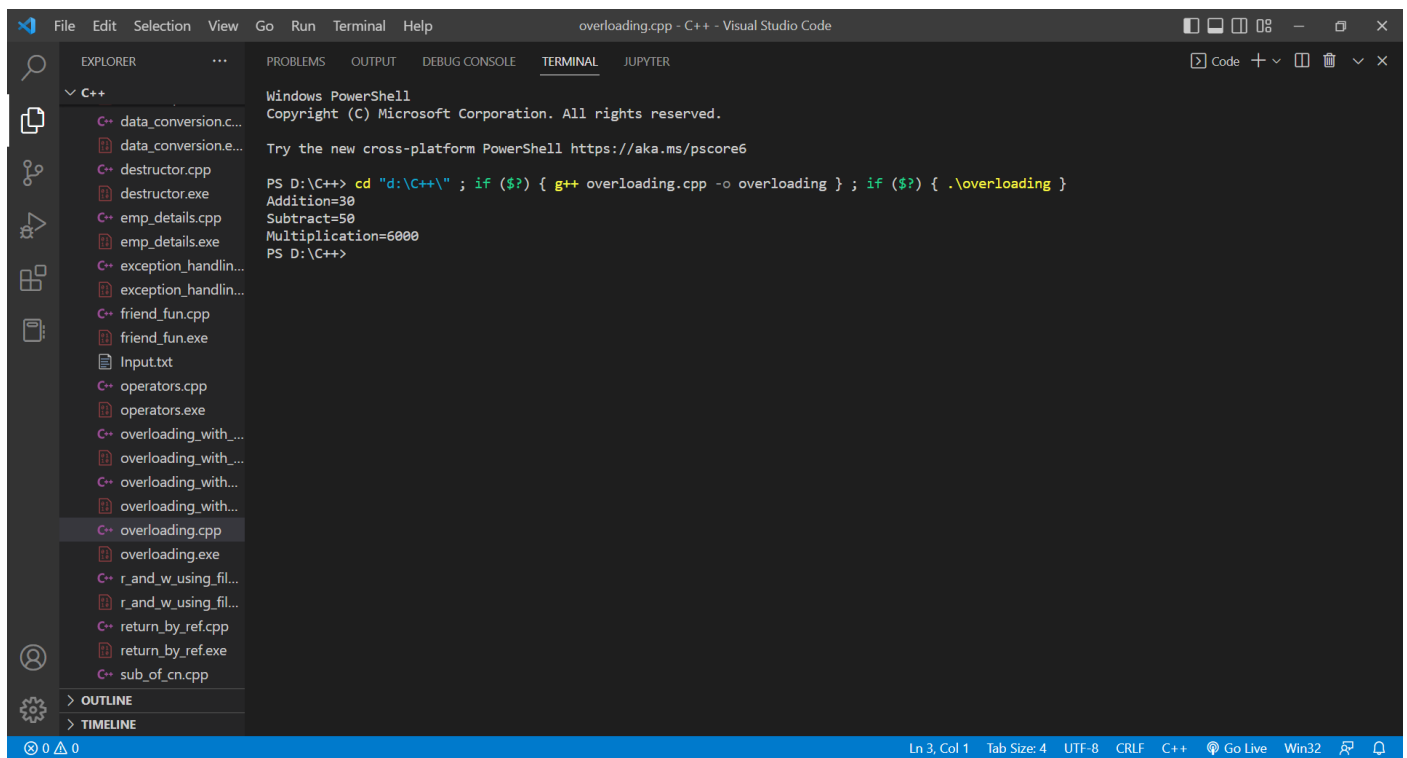
CODE: -

```
#include <iostream>
using namespace std;

class A{
    int a=10;
    int b=20;
public:
    void fun(){
        int sum=a+b;
        cout<<"Addition="<<sum<<endl;
    }
    void fun(int x,int y){
        int sub=x-y;
        cout<<"Subtract="<<sub<<endl;
    }
    void fun(int i,int j, int k){
        int mul=i*j*k;
        cout<<"Multiplication="<<mul<<endl;
    }
};

int main(){
    A a1;
    a1.fun();
    a1.fun(100,50);
    a1.fun(10,20,30);
    return 0;
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab active. The terminal window displays the output of a C++ compilation and execution process. The file explorer on the left shows a project named 'C++' with various source files and executables. The terminal output is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ overloading.cpp -o overloading } ; if ($?) { .\overloading }
Addition=30
Subtract=50
Multiplication=6000
PS D:\C++>
```

The file explorer on the left lists the following files and executables:

- data_conversion.c...
- data_conversion.e...
- destructor.cpp
- destructor.exe
- emp_details.cpp
- emp_details.exe
- exception_handlin...
- exception_handlin...
- friend_fun.cpp
- friend_fun.exe
- Input.txt
- operators.cpp
- operators.exe
- overloading_with_...
- overloading_with_...
- overloading_with...
- overloading_with...
- overloading.cpp
- overloading.exe
- r_and_w_using_fil...
- r_and_w_using_fil...
- return_by_ref.cpp
- return_by_ref.exe
- sub_of_cn.cpp

The status bar at the bottom indicates the current line and column (Ln 3, Col 1), tab size (4), encoding (UTF-8), line ending (CRLF), and the active language (C++).

5. WAP to implement Parameterized Constructor, Copy Constructor and Destructor.

CODE: -

i.

```
// CPP program to illustrate
// parameterized constructors
#include <iostream>
using namespace std;

class Point {
private:
    int x, y;

public:
    // Parameterized Constructor
    Point(int x1, int y1)
    {
        x = x1;
        y = y1;
    }

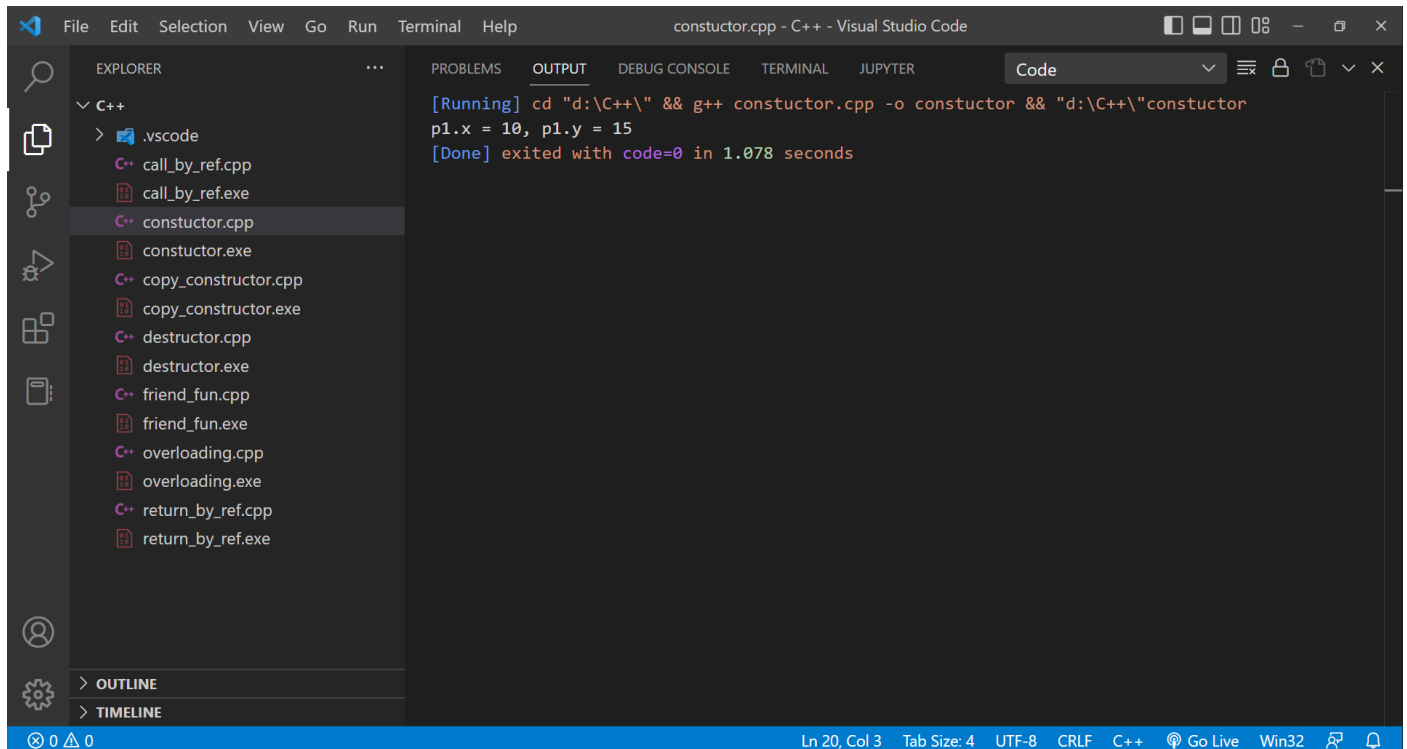
    int getX() { return x; }
    int getY() { return y; }
};

int main()
{
    // Constructor called
    Point p1(10, 15);

    // Access values assigned by constructor
    cout << "p1.x = " << p1.getX()
        << ", p1.y = " << p1.getY();

    return 0;
}
```

OUTPUT: -



The screenshot displays the Visual Studio Code interface with the 'constructor.cpp' file open. The Explorer sidebar on the left shows a project structure with files like `call_by_ref.cpp`, `call_by_ref.exe`, `constructor.cpp`, `constructor.exe`, `copy_constructor.cpp`, `copy_constructor.exe`, `destructor.cpp`, `destructor.exe`, `friend_fun.cpp`, `friend_fun.exe`, `overloading.cpp`, `overloading.exe`, `return_by_ref.cpp`, and `return_by_ref.exe`. The Output window on the right shows the execution command and its result:

```
[Running] cd "d:\C++\" && g++ constructor.cpp -o constructor && "d:\C++\"constructor
p1.x = 10, p1.y = 15
[Done] exited with code=0 in 1.078 seconds
```

The status bar at the bottom indicates the current line and column (Ln 20, Col 3), tab size (4), encoding (UTF-8), line ending (CRLF), and the active language (C++).

ii.

CODE: -

```
#include<iostream>
#include<string.h>
using namespace std;
class student
{
    int rno;
    char name[50];
    double fee;
public:
    student(int,char[],double);
    student(student &t) //copy constructor
    {
        rno=t.rno;
        strcpy(name,t.name);
        fee=t.fee;
    }
    void display();
};

student::student(int no,char n[],double f)
{
    rno=no;
    strcpy(name,n);
    fee=f;
}

void student::display()
{
    cout<<endl<<rno<<"\t"<<name<<"\t"<<fee;
}

int main()
{
    student s(1001,"Manjeet",10000);
    s.display();

    student manjeet(s); //copy constructor called
    manjeet.display();

    return 0;
}
```

OUTPUT: -

```
File Edit Selection View Go Run Terminal Help
copy_constructor.cpp - C++ - Visual Studio Code

EXPLORER
C++
  .vscode
  call_by_ref.cpp
  call_by_ref.exe
  constructor.cpp
  constructor.exe
  copy_constructor.cpp
  copy_constructor.exe
  destructor.cpp
  destructor.exe
  friend_fun.cpp
  friend_fun.exe
  overloading.cpp
  overloading.exe
  return_by_ref.cpp
  return_by_ref.exe

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
[Running] cd "d:\C++\" && g++ copy_constructor.cpp -o copy_constructor && "d:\C++\"copy_constructor
copy_constructor.cpp: In function 'int main()':
copy_constructor.cpp:38:32: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
    student s(1001,"Manjeet",10000);
                           ^
1001   Manjeet 10000
1001   Manjeet 10000
[Done] exited with code=0 in 1.092 seconds

Ln 12, Col 6 Tab Size: 4 UTF-8 CRLF C++ Go Live Win32
```

iii.

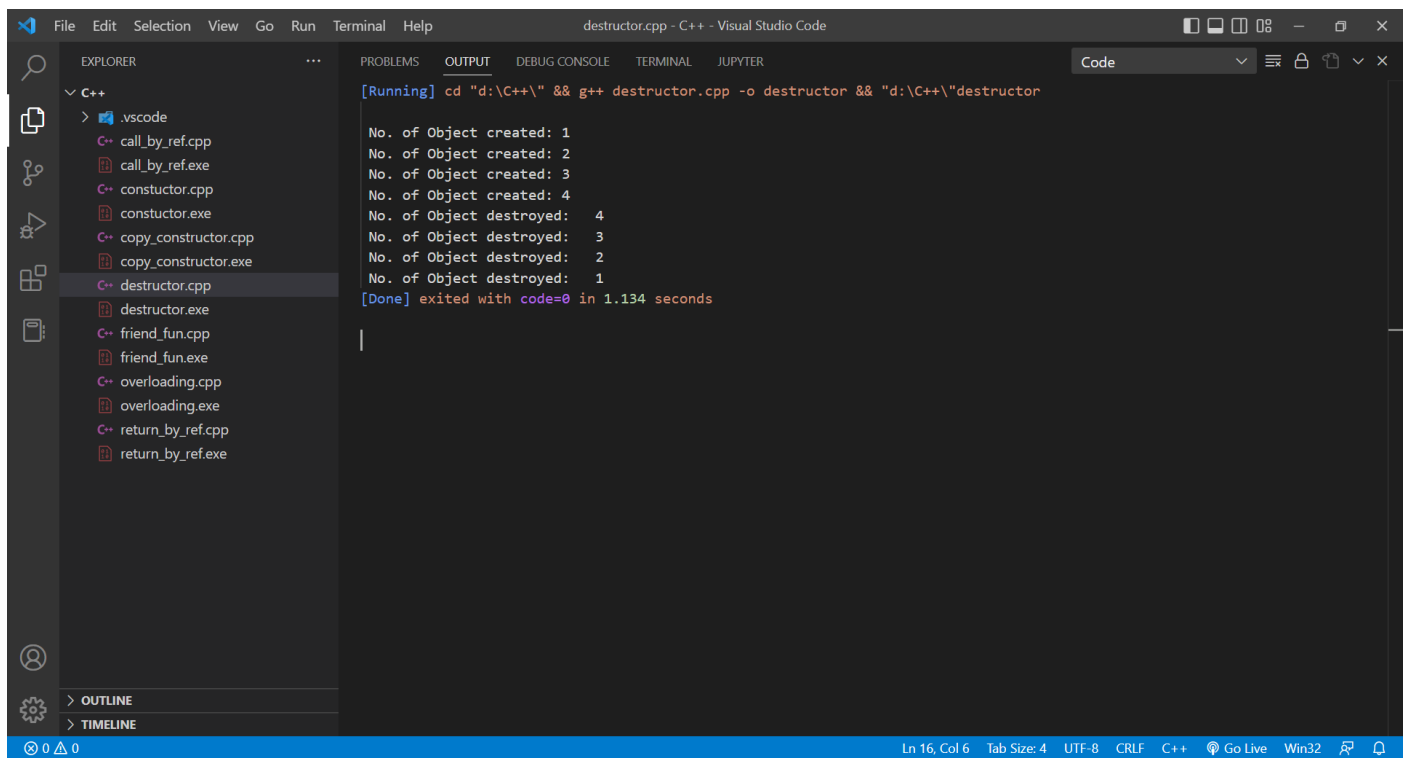
CODE: -

```
#include <iostream>
using namespace std;
int count = 0;
class Test {
public:
    Test()
    {
        count++;
        cout << "\n No. of Object created:\t" << count;
    }

    ~Test()
    {
        cout << "\n No. of Object destroyed:\t" << count;
        --count;
    }
};

main()
{
    Test t, t1, t2, t3;
    return 0;
}
```


OUTPUT: -



```
File Edit Selection View Go Run Terminal Help
destructor.cpp - C++ - Visual Studio Code

EXPLORER
C++
  .vscode
  call_by_ref.cpp
  call_by_ref.exe
  constructor.cpp
  constructor.exe
  copy_constructor.cpp
  copy_constructor.exe
  destructor.cpp
  destructor.exe
  friend_fun.cpp
  friend_fun.exe
  overloading.cpp
  overloading.exe
  return_by_ref.cpp
  return_by_ref.exe

OUTLINE
TIMELINE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
[Running] cd "d:\C++\" && g++ destructor.cpp -o destructor && "d:\C++\"destructor

No. of Object created: 1
No. of Object created: 2
No. of Object created: 3
No. of Object created: 4
No. of Object destroyed: 4
No. of Object destroyed: 3
No. of Object destroyed: 2
No. of Object destroyed: 1
[Done] exited with code=0 in 1.134 seconds

Ln 16, Col 6 Tab Size: 4 UTF-8 CRLF C++ Go Live Win32
```

6. WAP to show the usage of constructor in base and derived classes, in multiple inheritance.

CODE: -

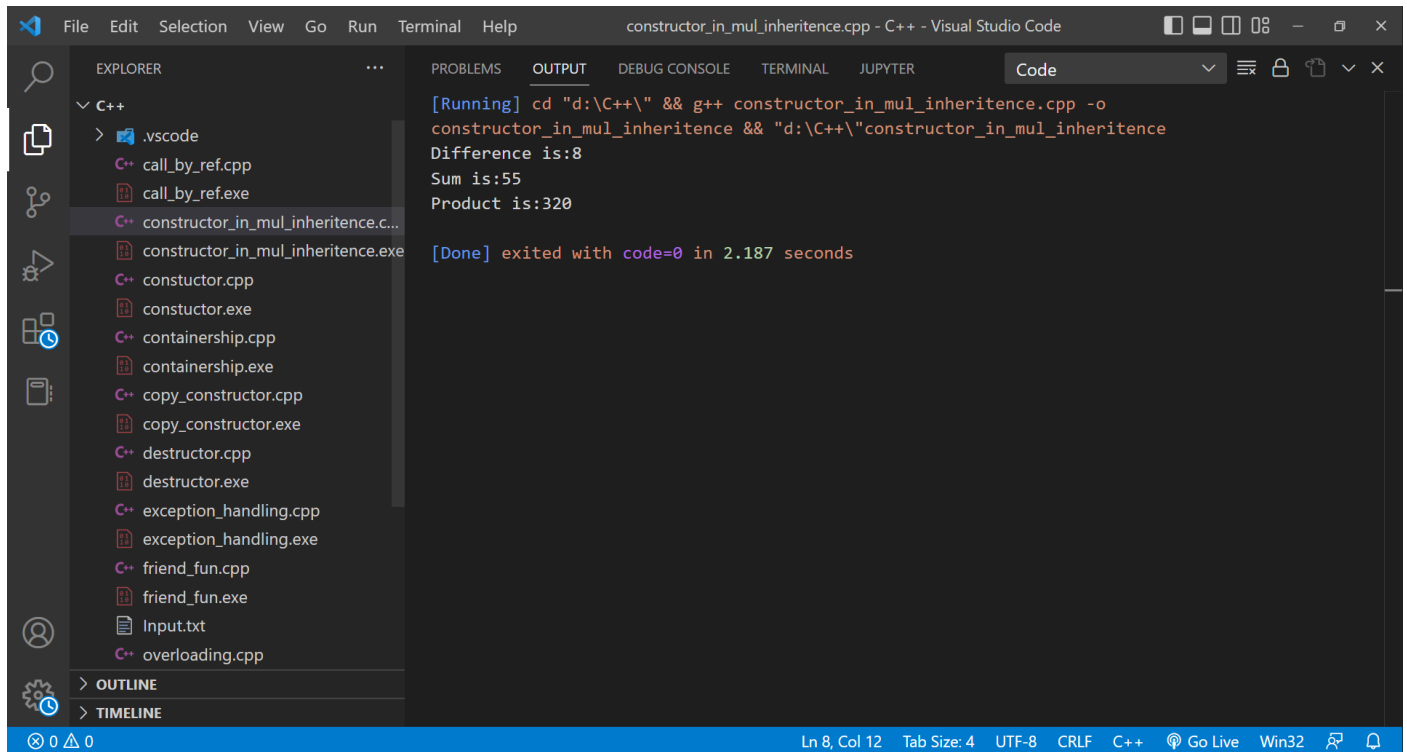
```
// C++ program to implement
// constructors in multiple
// inheritance
#include<iostream>
using namespace std;
class A1
{
    public:
        A1()
        {
            int a = 20, b = 35, c;
            c = a + b;
            cout << "Sum is:" <<
                c << endl;
        }
};

class A2
{
    public:
        A2()
        {
            int x = 50, y = 42, z;
            z = x - y;
            cout << "Difference is:" <<
                z << endl;
        }
};

class S: public A1,virtual A2
{
    public:
        S(): A1(), A2()
        {
            int r = 40, s = 8, t;
            t = r * s;
            cout << "Product is:" <<
                t << endl;
        }
};

// Driver code
int main()
{
    S obj;
    return 0;
}
```

OUTPUT: -



```
File Edit Selection View Go Run Terminal Help constructor_in_mul_inheritance.cpp - C++ - Visual Studio Code
```

EXPLORER

- C++
 - .vscode
 - call_by_ref.cpp
 - call_by_ref.exe
 - constructor_in_mul_inheritance.c...
 - constructor_in_mul_inheritance.exe
 - constructor.cpp
 - constructor.exe
 - containership.cpp
 - containership.exe
 - copy_constructor.cpp
 - copy_constructor.exe
 - destructor.cpp
 - destructor.exe
 - exception_handling.cpp
 - exception_handling.exe
 - friend_fun.cpp
 - friend_fun.exe
 - Input.txt
 - overloading.cpp
- OUTLINE
- TIMELINE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER Code

```
[Running] cd "d:\C++\" && g++ constructor_in_mul_inheritance.cpp -o
constructor_in_mul_inheritance && "d:\C++\"constructor_in_mul_inheritance
Difference is:8
Sum is:55
Product is:320

[Done] exited with code=0 in 2.187 seconds
```

Ln 8, Col 12 Tab Size: 4 UTF-8 CRLF C++ Go Live Win32

7. WAP to show the implementation of ‘containership’.

CODE: -

```
// CPP program to illustrate
// concept of Containership

#include <iostream>
using namespace std;

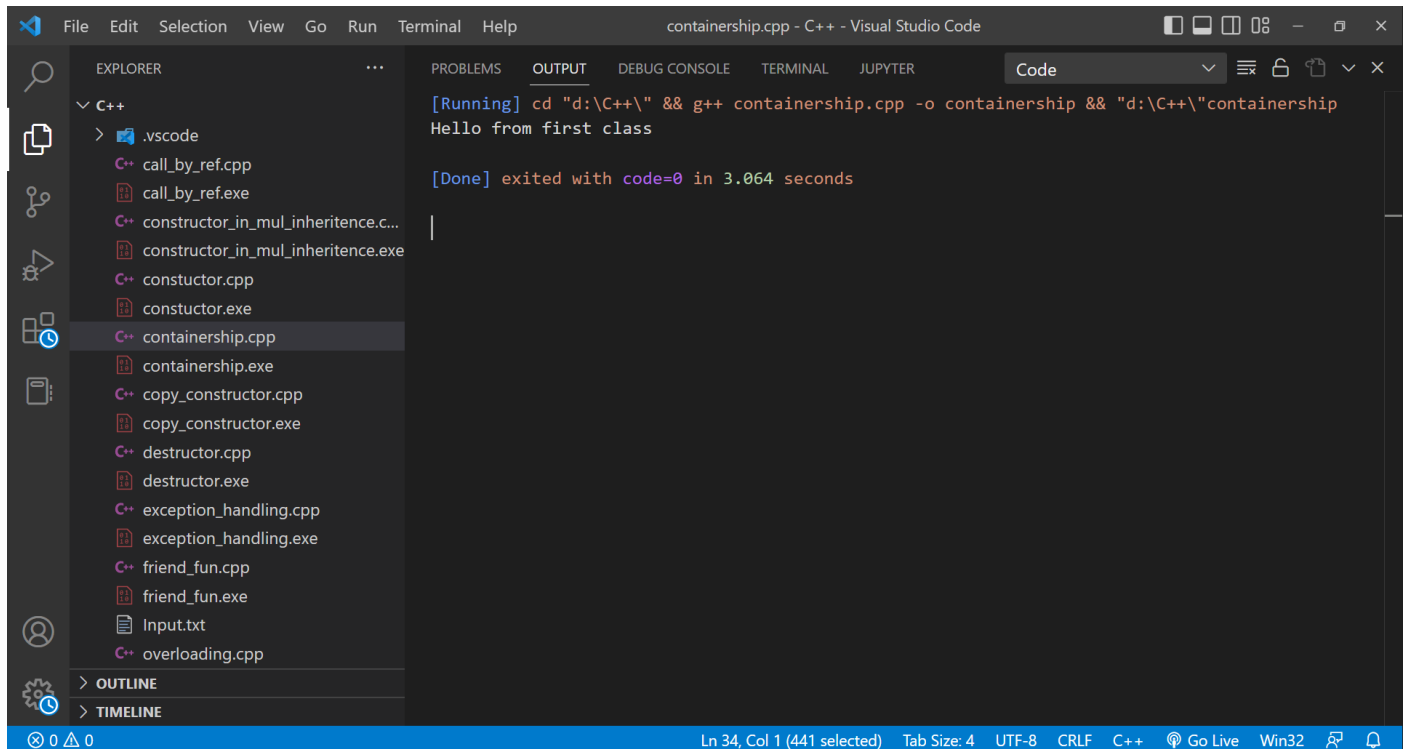
class first {
public:
    void showf()
    {
        cout << "Hello from first class\n";
    }
};

// Container class
class second {
    // creating object of first
    first f;

public:
    // constructor
    second()
    {
        // calling function of first class
        f.showf();
    }
};

int main()
{
    // creating object of second
    second s;
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'OUTPUT' panel active. The Explorer panel on the left shows a project named 'C++' with various files and executables. The 'containership.cpp' file is selected. The OUTPUT panel displays the following text:

```
[Running] cd "d:\C++\" && g++ containership.cpp -o containership && "d:\C++\"containership
Hello from first class

[Done] exited with code=0 in 3.064 seconds
```

The status bar at the bottom indicates the current line and column as 'Ln 34, Col 1 (441 selected)'.

8. WAP to show swapping using template function (Generic).

CODE: -

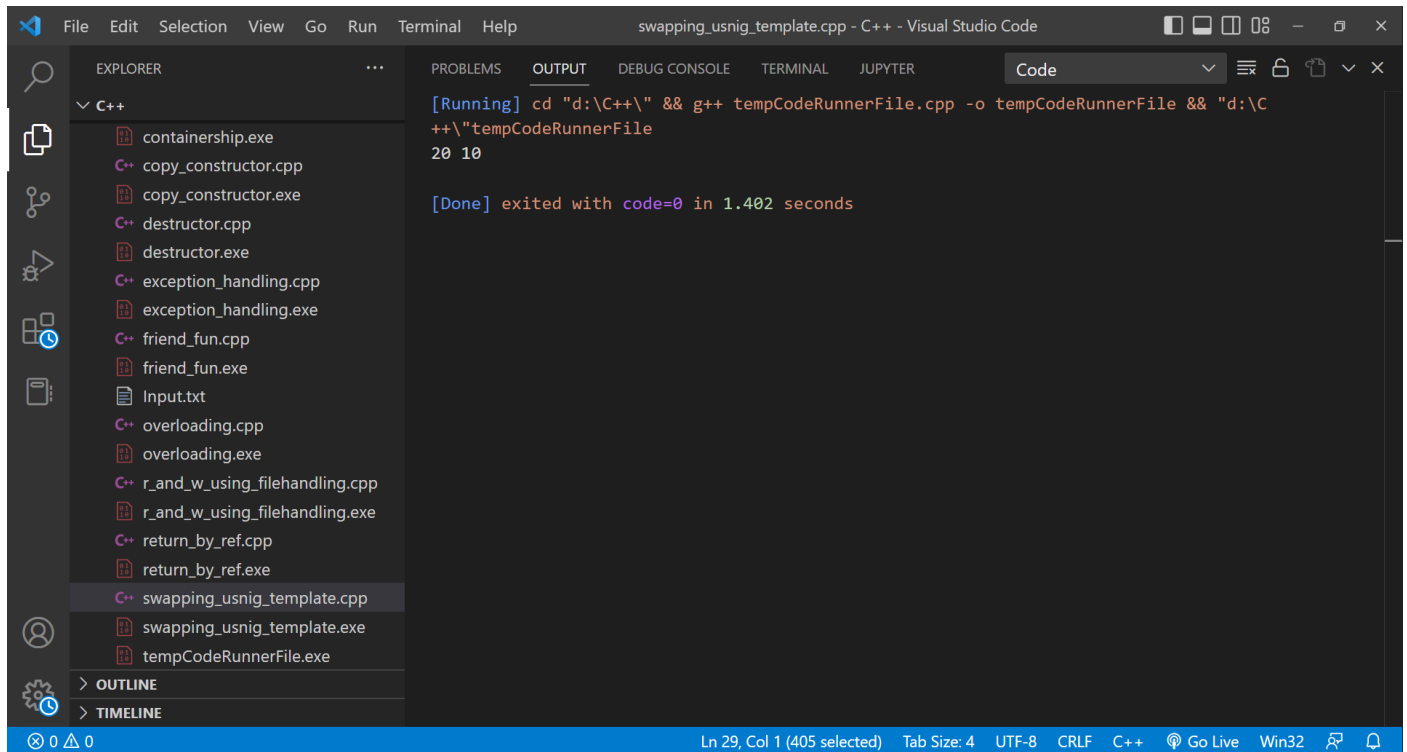
```
// C++ program to implement
// function templates
#include <iostream>
using namespace std;

// Function template to swap
// two numbers
template <class T>
int swap_numbers(T& x, T& y)
{
    T t;
    t = x;
    x = y;
    y = t;
    return 0;
}

// Driver code
int main()
{
    int a, b;
    a = 10, b = 20;

    // Invoking the swap()
    swap_numbers(a, b);
    cout << a << " " << b << endl;
    return 0;
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'OUTPUT' panel active. The Explorer panel on the left lists various C++ files and executables, with 'swapping_usnig_template.cpp' selected. The OUTPUT panel displays the execution output of the selected file, showing a successful run with a return code of 0.

```
swapping_usnig_template.cpp - C++ - Visual Studio Code
```

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER Code

[Running] cd "d:\C++\" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "d:\C++\tempCodeRunnerFile.exe"

20 10

[Done] exited with code=0 in 1.402 seconds

Ln 29, Col 1 (405 selected) Tab Size: 4 UTF-8 CRLF C++ Go Live Win32

9. WAP to implement 'Exception Handling'.

CODE: -

```
// C++ program to illustrate the concept  
// of exception handling using class
```

```
#include <bits/stdc++.h>  
using namespace std;
```

```
// Class declaration
```

```
class Number {
```

```
private:
```

```
    int a, b;
```

```
public:
```

```
    // Constructors
```

```
    Number(int x, int y)
```

```
{
```

```
    a = x;
```

```
    b = y;
```

```
}
```

```
// Function that find the GCD
```

```
// of two numbers a and b
```

```
int gcd()
```

```
{
```

```
    // While a is not equal to b
```

```
    while (a != b) {
```

```
        // Update a to a - b
```

```
        if (a > b)
```

```
            a = a - b;
```

```
        // Otherwise, update b
```

```
        else
```

```
            b = b - a;
```

```
    }
```

```
    // Return the resultant GCD
```

```
    return a;
```

```
}
```

```
// Function to check if the
```

```
// given number is prime
```

```
bool isPrime(int n)
```

```
{
```

```
    // Base Case
```

```
    if (n <= 1)
```

```
        return false;
```

```
    // Iterate over the range [2, N]
```



```

    for (int i = 2; i < n; i++) {

        // If n has more than 2
        // factors, then return
        // false
        if (n % i == 0)
            return false;
    }

    // Return true
    return true;
}

};

// Empty class
class MyPrimeException {
};

// Driver Code
int main()
{
    int x = 13, y = 56;

    Number num1(x, y);

    // Print the GCD of X and Y
    cout << "GCD is = "
        << num1.gcd() << endl;

    // If X is prime
    if (num1.isPrime(x))
        cout << x
            << " is a prime number"
            << endl;

    // If Y is prime
    if (num1.isPrime(y))
        cout << y
            << " is a prime number"
            << endl;

    // Exception Handling
    if ((num1.isPrime(x))
        || (num1.isPrime(y))) {

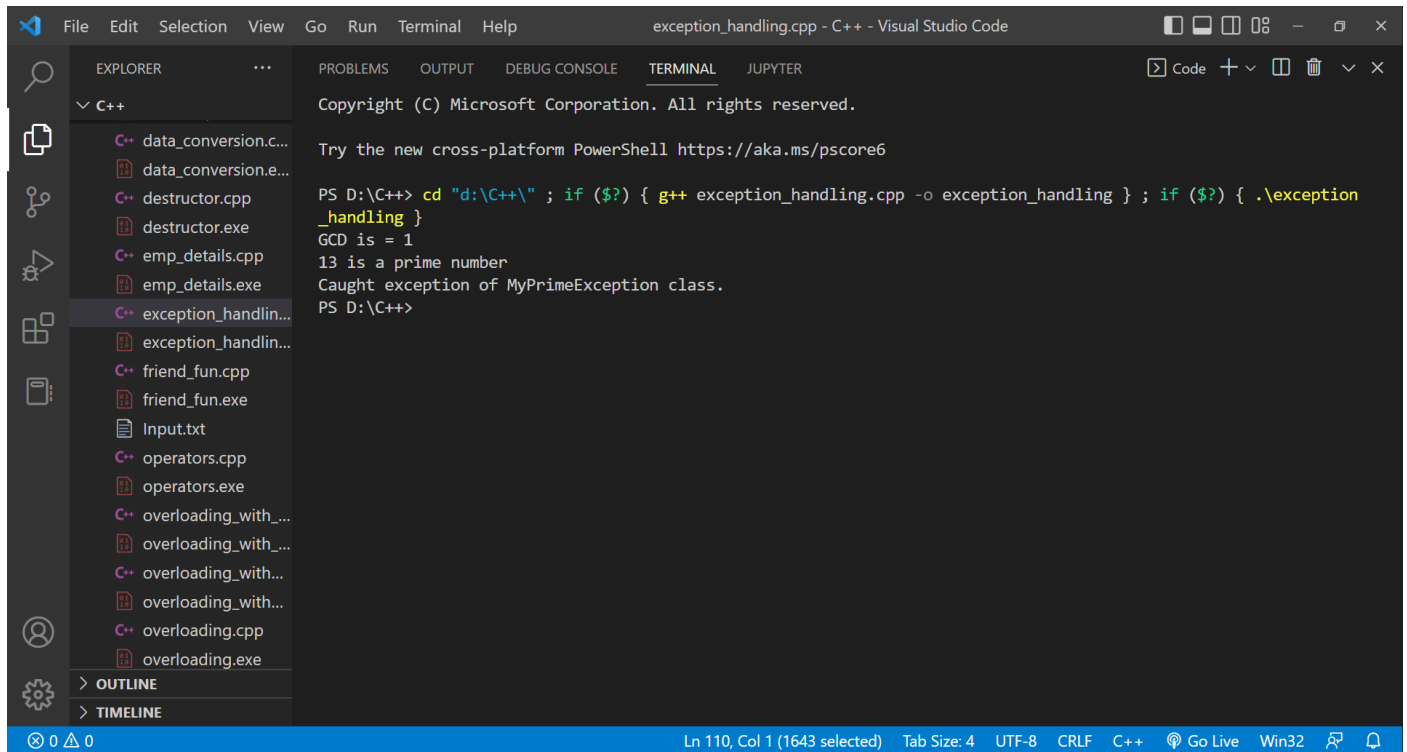
        // Try Block
        try {
            throw MyPrimeException();
        }

        // Catch Block

```

```
    catch (MyPrimeException t) {  
  
        cout << "Caught exception "  
            << "of MyPrimeException "  
            << "class." << endl;  
    }  
}  
  
return 0;  
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'exception_handling.cpp' file open. The 'TERMINAL' tab is active, displaying the output of a PowerShell command. The Explorer sidebar on the left lists various C++ files and executables, including 'data_conversion.c...', 'destructor.cpp', 'emp_details.cpp', 'exception_handling...', 'friend_fun.cpp', 'operators.cpp', 'overloading_with...', and 'overloading.cpp'. The terminal output shows the execution of a C++ program that demonstrates exception handling, including a GCD calculation and a prime number check.

```
exception_handling.cpp - C++ - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
C++
  C++ data_conversion.c...
  data_conversion.e...
  C++ destructor.cpp
  destructor.exe
  C++ emp_details.cpp
  emp_details.exe
  C++ exception_handlin...
  exception_handlin...
  C++ friend_fun.cpp
  friend_fun.exe
  Input.txt
  C++ operators.cpp
  operators.exe
  C++ overloading_with_...
  overloading_with_...
  C++ overloading_with...
  overloading_with...
  C++ overloading.cpp
  overloading.exe
  > OUTLINE
  > TIMELINE
  0 0 0
  Ln 110, Col 1 (1643 selected) Tab Size: 4 UTF-8 CRLF C++ Go Live Win32

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ exception_handling.cpp -o exception_handling } ; if ($?) { .\exception_handling }
GCD is = 1
13 is a prime number
Caught exception of MyPrimeException class.
PS D:\C++>
```

10.WAP to read and write values through object using file handling.

CODE: -

```
/* File Handling with C++ using ifstream & ofstream class object*/
/* To write the Content in File*/
/* Then to read the content of file*/
#include <iostream>

/* fstream header file for ifstream, ofstream,
fstream classes */
#include <fstream>

using namespace std;

// Driver Code
int main()
{
    // Creation of ofstream class object
    ofstream fout;

    string line;

    // by default ios::out mode, automatically deletes
    // the content of file. To append the content, open in ios::app
    // fout.open("sample.txt", ios::app)
    fout.open("text.txt");

    // Execute a loop If file successfully opened
    while (fout) {

        // Read a Line from standard input
        getline(cin, line);

        // Press -1 to exit
        if (line == "-1")
            break;

        // Write line in file
        fout << line << endl;
    }

    // Close the File
    fout.close();

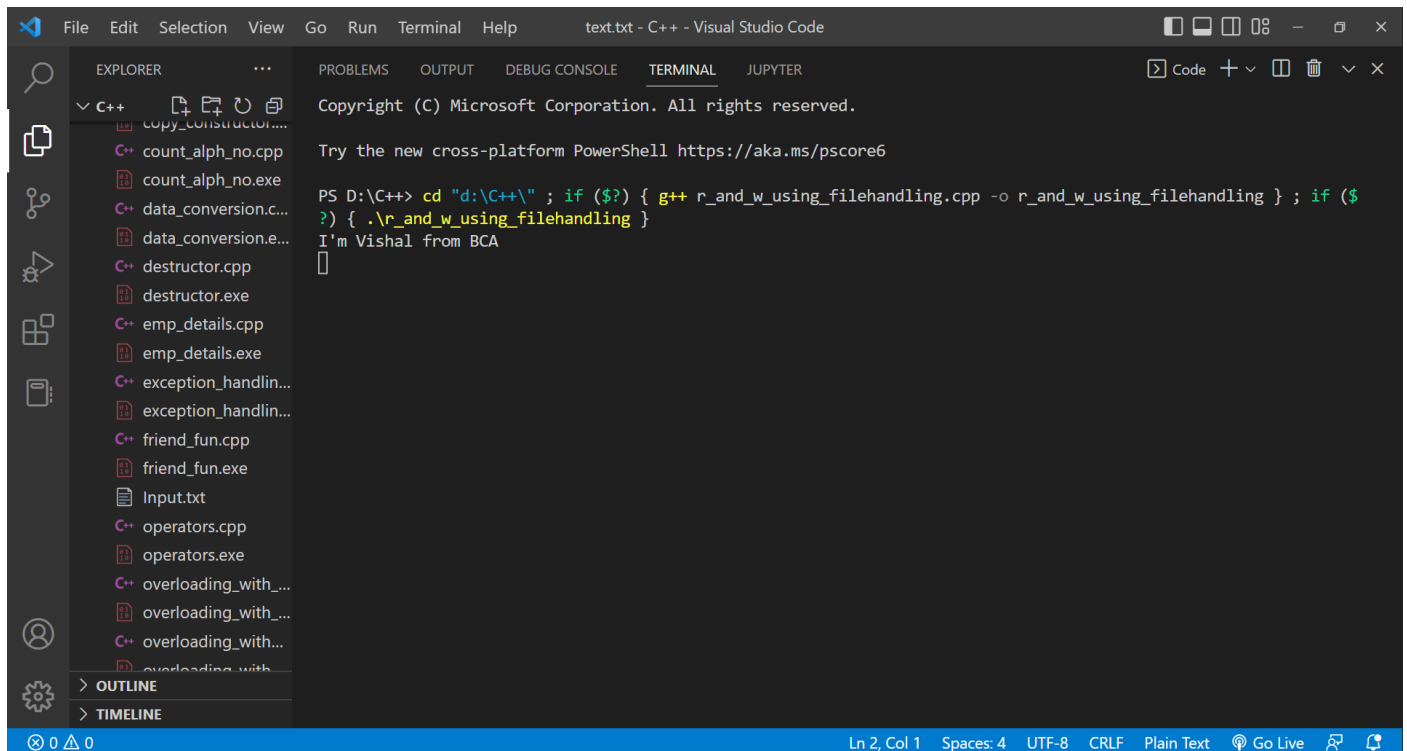
    // Creation of ifstream class object to read the file
    ifstream fin;

    // by default open mode = ios::in mode
    fin.open("text.txt");

    // Execute a loop until EOF (End of File)
```

```
while (getline(fin, line)) {  
  
    // Print line (read from file) in Console  
    cout << line << endl;  
}  
  
// Close the file  
fin.close();  
  
return 0;  
}
```

OUTPUT: -



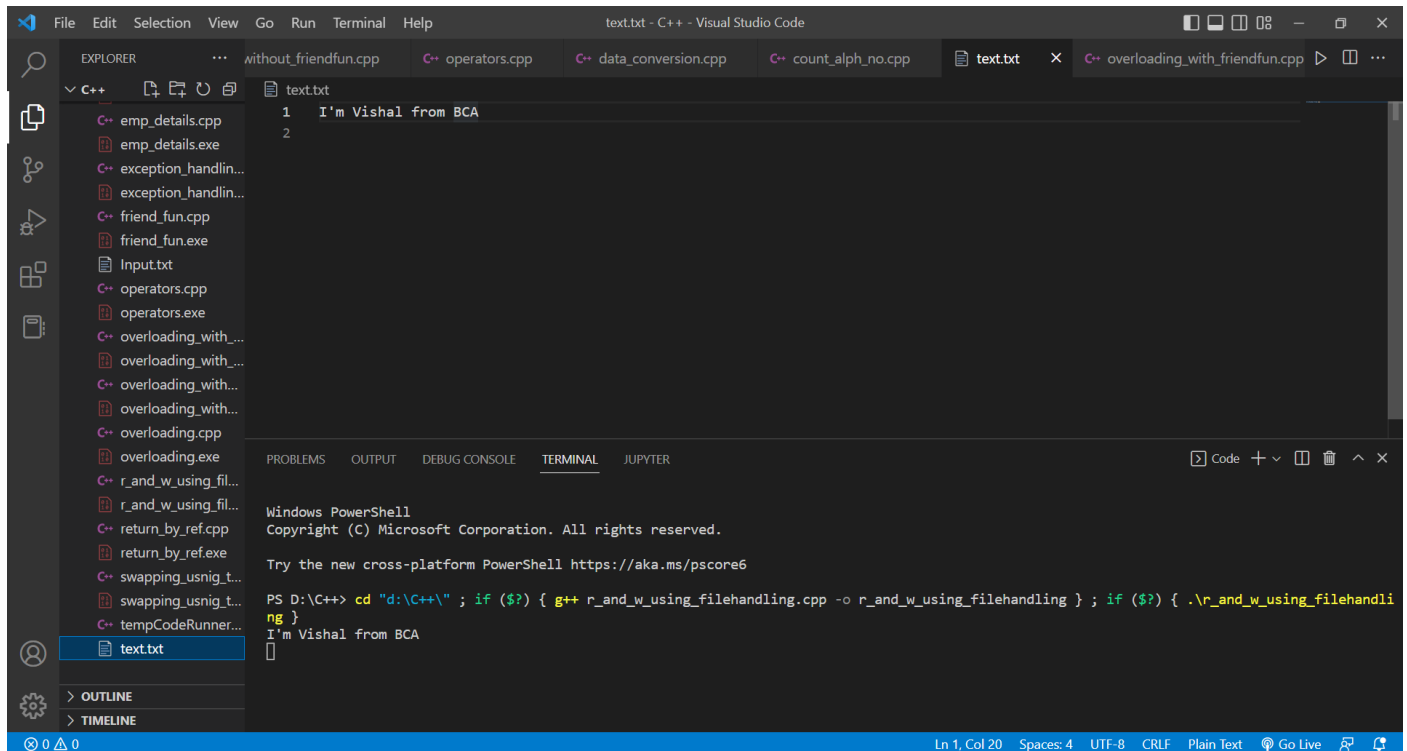
The screenshot shows the Visual Studio Code interface with the terminal panel active. The terminal displays the output of a C++ program. The output starts with a copyright notice for Microsoft Corporation, followed by a message about the new cross-platform PowerShell. Then, the program prints "I'm Vishal from BCA" and a blank line. The status bar at the bottom indicates the current line and column as Ln 2, Col 1.

```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ r_and_w_using_filehandling.cpp -o r_and_w_using_filehandling } ; if ($?) { .\r_and_w_using_filehandling }
I'm Vishal from BCA

```



The screenshot shows the Visual Studio Code interface with the Explorer panel active. The Explorer panel displays the file structure of the project, including the source code files and the output files. The file "text.txt" is selected, and its content is displayed in the editor. The content of "text.txt" is "I'm Vishal from BCA". The status bar at the bottom indicates the current line and column as Ln 1, Col 20.

```
1 I'm Vishal from BCA
2

```

11. Create a class employee which have name, age and address of employee, include functions getdata() and showdata(), getdata() takes the input from the user, showdata() display the data in following format:

Name:

Age:

Address:

CODE: -

```
#include <windows.h>
```

```
#include <iostream>
```

```
using namespace std;
```

```
class employee
```

```
{
```

```
    int emp_number;
```

```
    char emp_name[20];
```

```
    float emp_basic;
```

```
    float emp_da;
```

```
    float emp_it;
```

```
    float emp_net_sal;
```

```
    char emp_address[50];
```

```
public:
```

```
    void get_emp_details();
```

```
    float find_net_salary(float basic, float da, float it);
```

```
    void show_emp_details();
```

```
};
```

```
void employee :: get_emp_details()
```

```
{
```

```
    cout<<"\nEnter employee number: ";
```

```
    cin>>emp_number;
```

```
    cout<<"\nEnter employee name: ";
```

```
    cin>>emp_name;
```

```
    cout<<"\nEnter employee basic: ";
```

```
    cin>>emp_basic;
```

```
    cout<<"\nEnter employee DA: ";
```

```
    cin>>emp_da;
```

```
    cout<<"\nEnter employee IT: ";
```

```
    cin>>emp_it;
```

```
    cout<<"\nEnter employee address: ";
```

```
    cin>>emp_address;
```

```
}
```

```
float employee :: find_net_salary(float basic, float da, float it)
```

```
{
```

```
    return (basic+da)-it;
```

```

}

void employee :: show_emp_details()
{
    cout<<"\n\n**** Details of Employee ****";
    cout<<"\nEmployee Name    : "<<emp_name;
    cout<<"\nEmployee number  : "<<emp_number;
    cout<<"\nBasic salary    : "<<emp_basic;
    cout<<"\nEmployee DA      : "<<emp_da;
    cout<<"\nIncome Tax      : "<<emp_it;
    cout<<"\nEmployee address   : "<<emp_address;
    cout<<"\nNet Salary      : "<<find_net_salary(emp_basic, emp_da, emp_it);
    cout<<"\n-----\n\n";
}

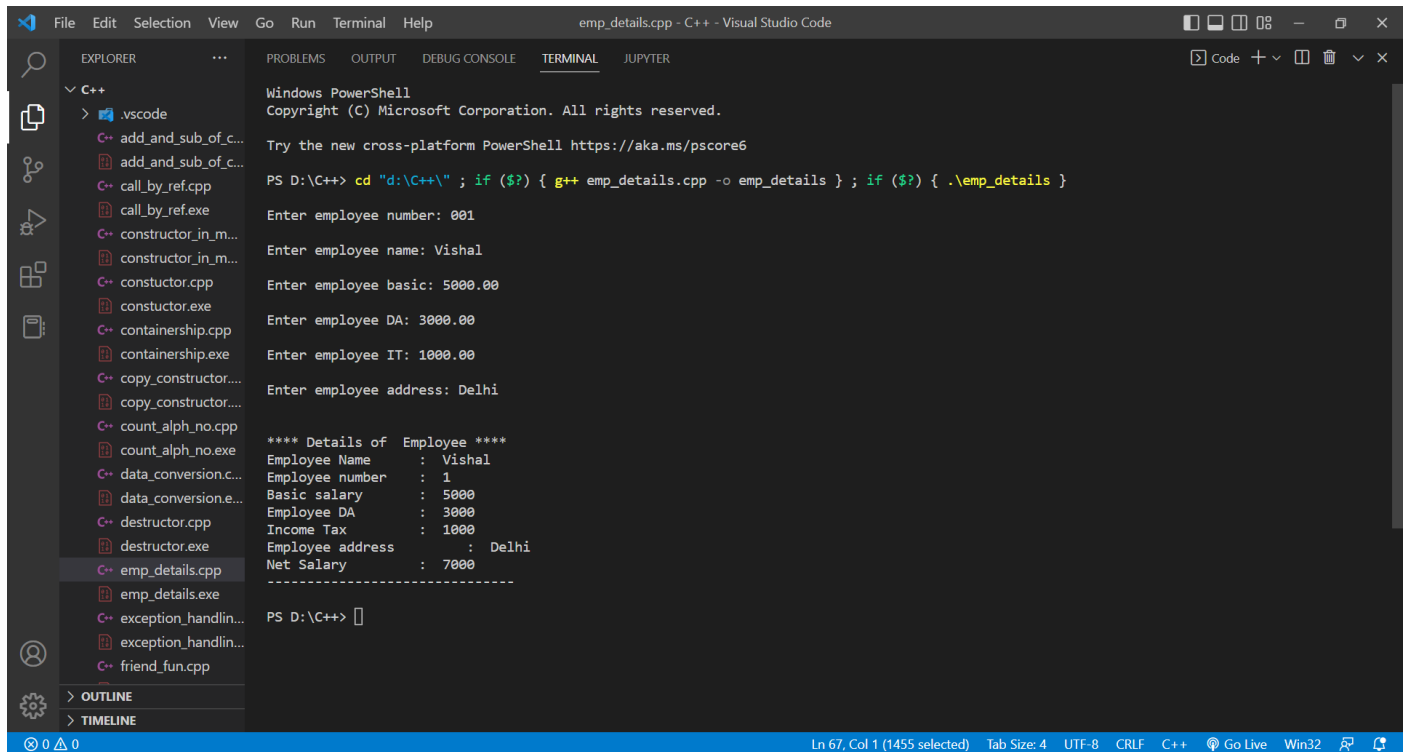
int main()
{
    employee emp;

    emp.get_emp_details();
    emp.show_emp_details();

    return 0;
}

```


OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab active. The terminal window displays the output of a C++ program named 'emp_details.exe'. The program prompts the user to enter employee details: number, name, basic salary, DA, IT, and address. After entering the details, it prints a formatted table of the employee's information.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ emp_details.cpp -o emp_details } ; if ($?) { .\emp_details }

Enter employee number: 001
Enter employee name: Vishal
Enter employee basic: 5000.00
Enter employee DA: 3000.00
Enter employee IT: 1000.00
Enter employee address: Delhi

**** Details of Employee ****
Employee Name      : Vishal
Employee number    : 1
Basic salary       : 5000
Employee DA        : 3000
Income Tax         : 1000
Employee address   : Delhi
Net Salary         : 7000
-----

PS D:\C++> 
```

The status bar at the bottom indicates the current line and column (Ln 67, Col 1), tab size (4), encoding (UTF-8), and line endings (CRLF).

12.WAP to add and subtract two complex numbers using classes.

CODE: -

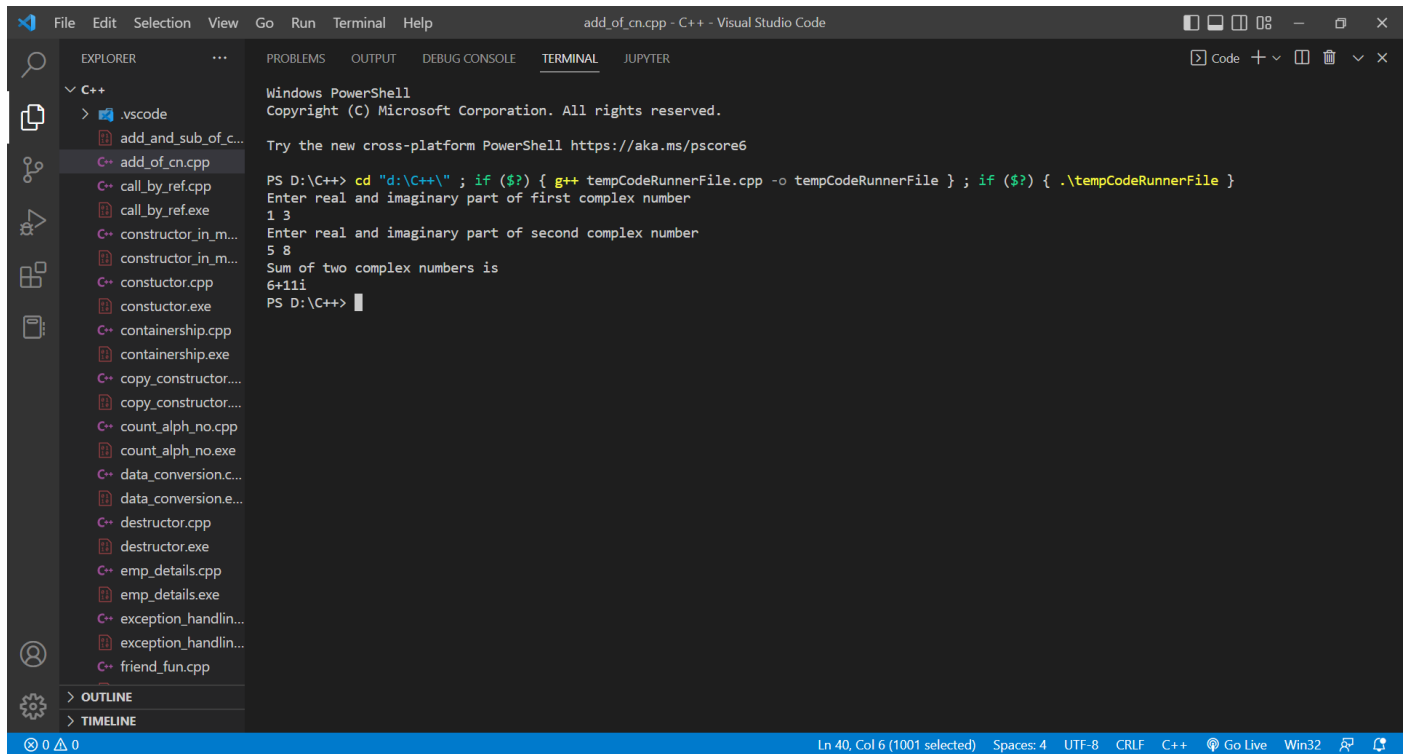
i.

```
/* C++ Program to add two Complex Numbers */
#include<iostream>
using namespace std;
class Complex{
public:
    int real;
    int imag;
    /* Function to set the values of
    * real and imaginary part of each complex number
    */
    void setvalue()
    {
        cin>>real;
        cin>>imag;
    }
    /* Function to display the sum of two complex numbers */
    void display()
    {
        cout<<real<<"+"<<imag<<"i"<<endl;
    }
    /* Function to add two complex numbers */

    void sum(Complex c1, Complex c2)
    {
        real=c1.real+c2.real;
        imag=c1.imag+c2.imag;
    }
};

int main()
{
    Complex c1,c2,c3;
    cout<<"Enter real and imaginary part of first complex number"<<endl;
    c1.setvalue();
    cout<<"Enter real and imaginary part of second complex number"<<endl;
    c2.setvalue();
    cout<<"Sum of two complex numbers is"<<endl;
    c3.sum(c1,c2);
    c3.display();
    return 0;
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab active. The terminal window displays the output of a C++ program. The program prompts the user to enter the real and imaginary parts of two complex numbers. The user enters '1 3' for the first number and '5 8' for the second number. The program then outputs the sum of the two complex numbers as '6+11i'.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter real and imaginary part of first complex number
1 3
Enter real and imaginary part of second complex number
5 8
Sum of two complex numbers is
6+11i
PS D:\C++>
```

The Explorer sidebar on the left shows a list of files in the 'C++' directory, including .vscode, add_and_sub_of_C..., add_of_cn.cpp, call_by_ref.cpp, call_by_ref.exe, constructor_in_m..., constructor_in_m..., constructor.cpp, constructor.exe, containership.cpp, containership.exe, copy_constructor..., copy_constructor..., count_alph_no.cpp, count_alph_no.exe, data_conversion.c..., data_conversion.e..., destructor.cpp, destructor.exe, emp_details.cpp, emp_details.exe, exception_handlin..., exception_handlin..., and friend_fun.cpp. The bottom status bar indicates 'Ln 40, Col 6 (1001 selected)' and 'Spaces: 4 UTF-8 CRLF C++ Go Live Win32'.

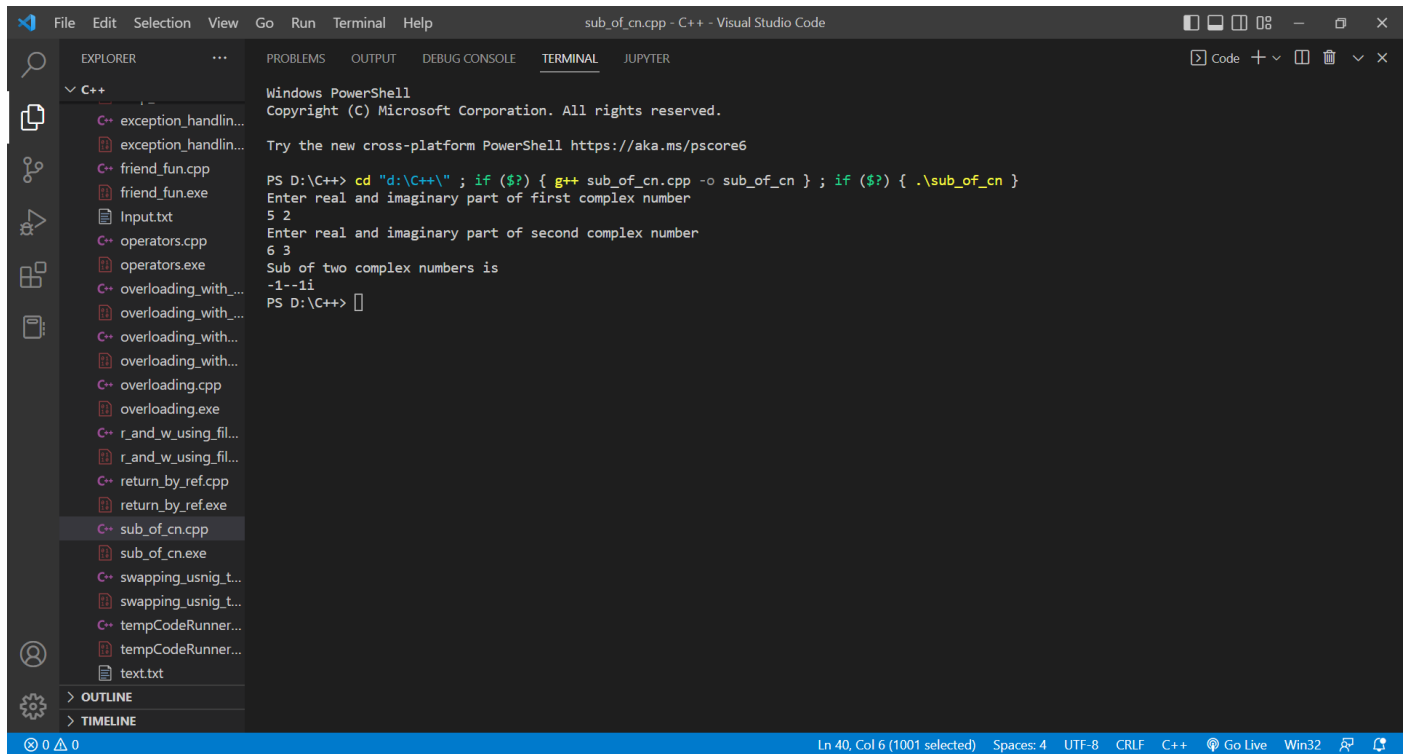
ii.

```
/* C++ Program to sub two Complex Numbers */
#include<iostream>
using namespace std;
class Complex{
public:
    int real;
    int imag;
    /* Function to set the values of
    * real and imaginary part of each complex number
    */
    void setvalue()
    {
        cin>>real;
        cin>>imag;
    }
    /* Function to display the sub of two complex numbers */
    void display()
    {
        cout<<real<<"-"<<imag<<"i"<<endl;
    }
    /* Function to sub two complex numbers */

    void sub(Complex c1, Complex c2)
    {
        real=c1.real-c2.real;
        imag=c1.imag-c2.imag;
    }
};

int main()
{
    Complex c1,c2,c3;
    cout<<"Enter real and imaginary part of first complex number"<<endl;
    c1.setvalue();
    cout<<"Enter real and imaginary part of second complex number"<<endl;
    c2.setvalue();
    cout<<"Sub of two complex numbers is"<<endl;
    c3.sub(c1,c2);
    c3.display();
    return 0;
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab active. The terminal window displays the output of a C++ program execution in a Windows PowerShell environment. The program, 'sub_of_cn.exe', prompts the user to enter the real and imaginary parts of two complex numbers. The user enters '5 2' for the first number and '6 3' for the second. The program then outputs the result: 'Sub of two complex numbers is -1--1i'.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ sub_of_cn.cpp -o sub_of_cn } ; if ($?) { .\sub_of_cn }
Enter real and imaginary part of first complex number
5 2
Enter real and imaginary part of second complex number
6 3
Sub of two complex numbers is
-1--1i
PS D:\C++> 
```

13. Write program to overload Binary + to add two similar types of objects. (Both with and without using friend functions).

CODE: -

i.

```
// C++ program to show binary operator overloading
#include <iostream>

using namespace std;

class Distance {
public:

    // Member Object
    int feet, inch;

    // No Parameter Constructor
    Distance()
    {
        this->feet = 0;
        this->inch = 0;
    }

    // Constructor to initialize the object's value
    // Parameterized Constructor
    Distance(int f, int i)
    {
        this->feet = f;
        this->inch = i;
    }

    // Declaring friend function using friend keyword
    friend Distance operator+(Distance&, Distance&);
};

// Implementing friend function with two parameters
Distance operator+(Distance& d1, Distance& d2) // Call by reference
{
    // Create an object to return
    Distance d3;

    // Perform addition of feet and inches
    d3.feet = d1.feet + d2.feet;
    d3.inch = d1.inch + d2.inch;

    // Return the resulting object
    return d3;
}

// Driver Code
```

```
int main()
{
    // Declaring and Initializing first object
    Distance d1(8, 9);

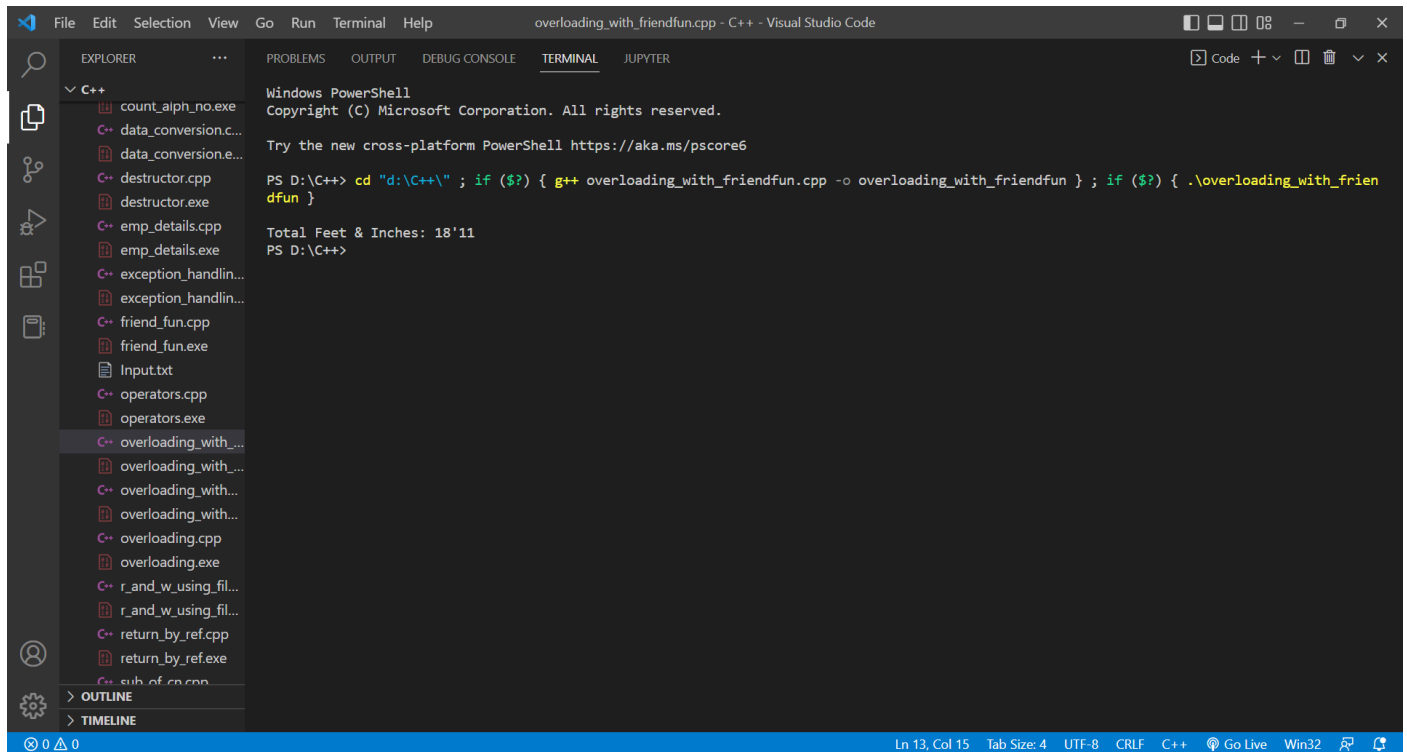
    // Declaring and Initializing second object
    Distance d2(10, 2);

    // Declaring third object
    Distance d3;

    // Use overloaded operator
    d3 = d1 + d2;

    // Display the result
    cout << "\nTotal Feet & Inches: " << d3.feet << " " << d3.inch;
    return 0;
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab active. The terminal window displays the output of a C++ program execution in a Windows PowerShell environment. The file explorer on the left shows a project named 'overloading_with_friendfun.cpp' with various source files and executables. The terminal output includes the Windows PowerShell prompt, copyright information, a command to run the program, and the program's output: 'Total Feet & Inches: 18'11'.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ overloading_with_friendfun.cpp -o overloading_with_friendfun } ; if ($?) { .\overloading_with_friendfun }

Total Feet & Inches: 18'11
PS D:\C++>
```


ii.

```
// C++ program to show binary operator overloading
#include <iostream>

using namespace std;

class Distance {
public:
    // Member Object
    int feet, inch;
    // No Parameter Constructor
    Distance()
    {
        this->feet = 0;
        this->inch = 0;
    }

    // Constructor to initialize the object's value
    // Parameterized Constructor
    Distance(int f, int i)
    {
        this->feet = f;
        this->inch = i;
    }

    // Overloading (+) operator to perform addition of
    // two distance object
    Distance operator+(Distance& d2) // Call by reference
    {
        // Create an object to return
        Distance d3;

        // Perform addition of feet and inches
        d3.feet = this->feet + d2.feet;
        d3.inch = this->inch + d2.inch;

        // Return the resulting object
        return d3;
    }
};

// Driver Code
int main()
{
    // Declaring and Initializing first object
    Distance d1(8, 9);

    // Declaring and Initializing second object
    Distance d2(10, 2);

    // Declaring third object
```

Distance d3;

// Use overloaded operator

d3 = d1 + d2;

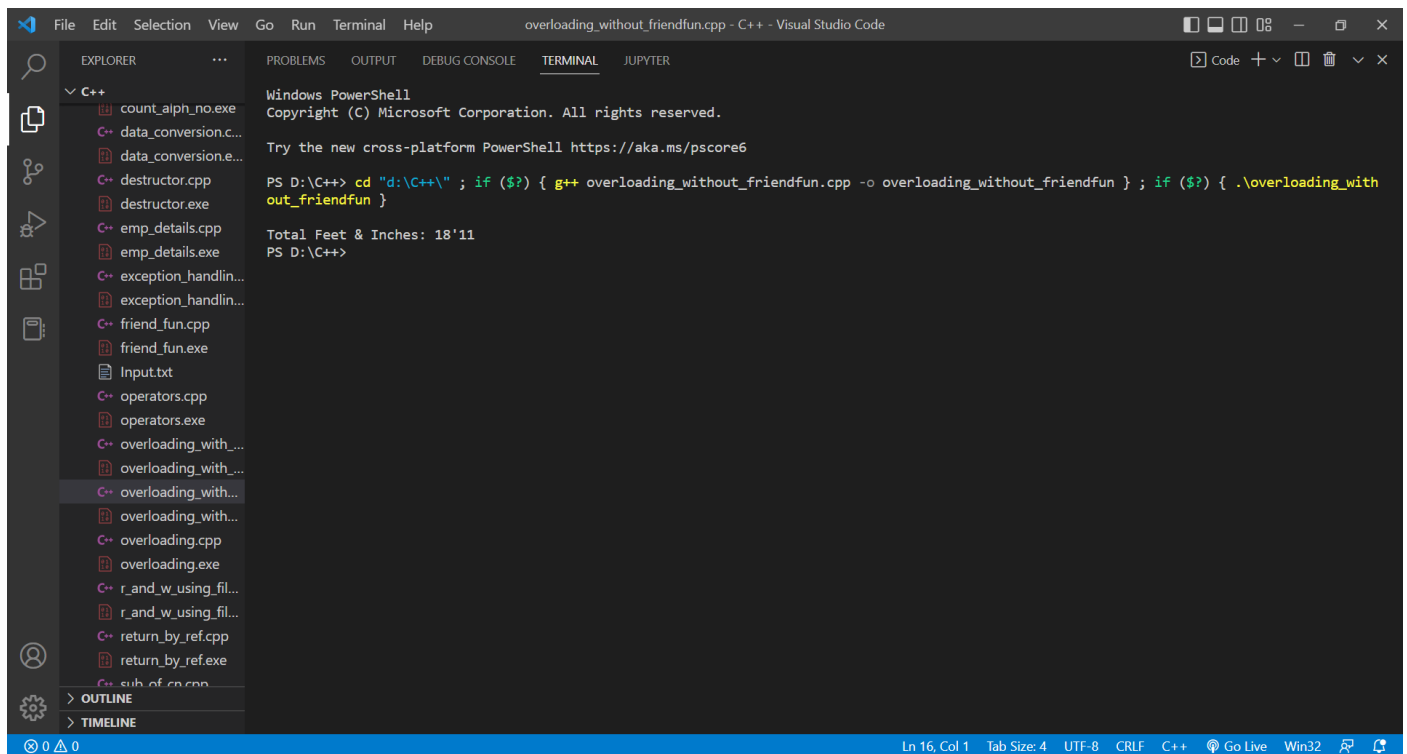
// Display the result

cout << "\nTotal Feet & Inches: " << d3.feet << " " << d3.inch;

return 0;

}

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab active. The terminal window is titled 'overloading_without_friendfun.cpp - C++ - Visual Studio Code'. The terminal output is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ overloading_without_friendfun.cpp -o overloading_without_friendfun } ; if ($?) { .\overloading_with_out_friendfun }

Total Feet & Inches: 18'11
PS D:\C++>
```

The Explorer sidebar on the left shows a list of files in the 'C++' directory, including 'count_alph_no.exe', 'data_conversion.c...', 'data_conversion.e...', 'destructor.cpp', 'destructor.exe', 'emp_details.cpp', 'emp_details.exe', 'exception_handlin...', 'exception_handlin...', 'friend_fun.cpp', 'friend_fun.exe', 'Input.txt', 'operators.cpp', 'operators.exe', 'overloading_with_...', 'overloading_with_...', 'overloading_with...', 'overloading_with...', 'overloading.cpp', 'overloading.exe', 'r_and_w_using_fil...', 'r_and_w_using_fil...', 'return_by_ref.cpp', 'return_by_ref.exe', and 'C++ sub of c++.cpp'. The 'OUTLINE' and 'TIMELINE' tabs are also visible at the bottom of the Explorer sidebar.

The status bar at the bottom indicates 'Ln 16, Col 1', 'Tab Size: 4', 'UTF-8', 'CRLF', 'C++', 'Go Live', 'Win32', and '0 0 0'.

14.WAP to implement += and = operator.

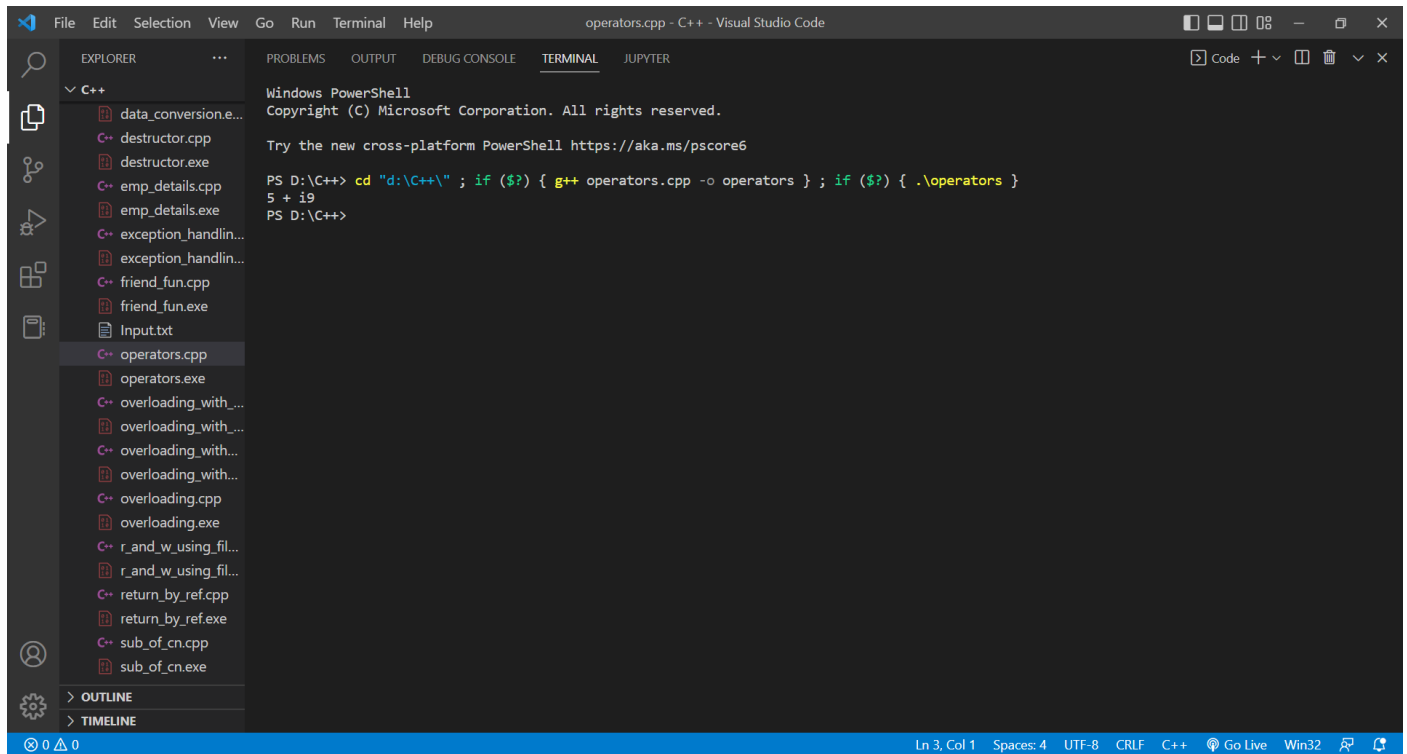
CODE: -

```
#include <iostream>
using namespace std;

class ComplexNumber{
private:
int real;
int imaginary;
public:
ComplexNumber(int real, int imaginary){
    this->real = real;
    this->imaginary = imaginary;
}
void print(){
    cout<<real<<" + i"<<imaginary;
}
ComplexNumber operator+ (ComplexNumber c2){
    ComplexNumber c3(0,0);
    c3.real = this->real+c2.real;
    c3.imaginary = this->imaginary + c2.imaginary;
    return c3;
}
};

int main() {
    ComplexNumber c1(3,5);
    ComplexNumber c2(2,4);
    ComplexNumber c3 = c1 + c2;
    c3.print();
    return 0;
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab active. The terminal window displays the output of a C++ compilation and execution process. The file explorer on the left shows a project structure with various C++ files, including 'operators.cpp' which is currently selected. The terminal output shows the PowerShell prompt, the directory change to 'D:\C++', the compilation command using 'g++', and the successful execution of the 'operators.exe' program, which outputs '5 + i9'.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ operators.cpp -o operators } ; if ($?) { .\operators }
5 + i9
PS D:\C++>
```

15.WAP to convert meter to centimetre and vice versa, using data conversions and operator overloading.

CODE: -

```
// C++ program to illustrate the
// above conversion
#include <bits/stdc++.h>
using namespace std;
//minutes class
class Minute {

public:
int mins;
    // Constructors
    Minute()
    {
        mins = 0;
    }

    // Function to print the value of
    // hours and minutes
    void show()
    {
        cout << "\nTotal Minute : " << mins << endl;
    }
};

// Time Class
class Time {
    int hr, mins;

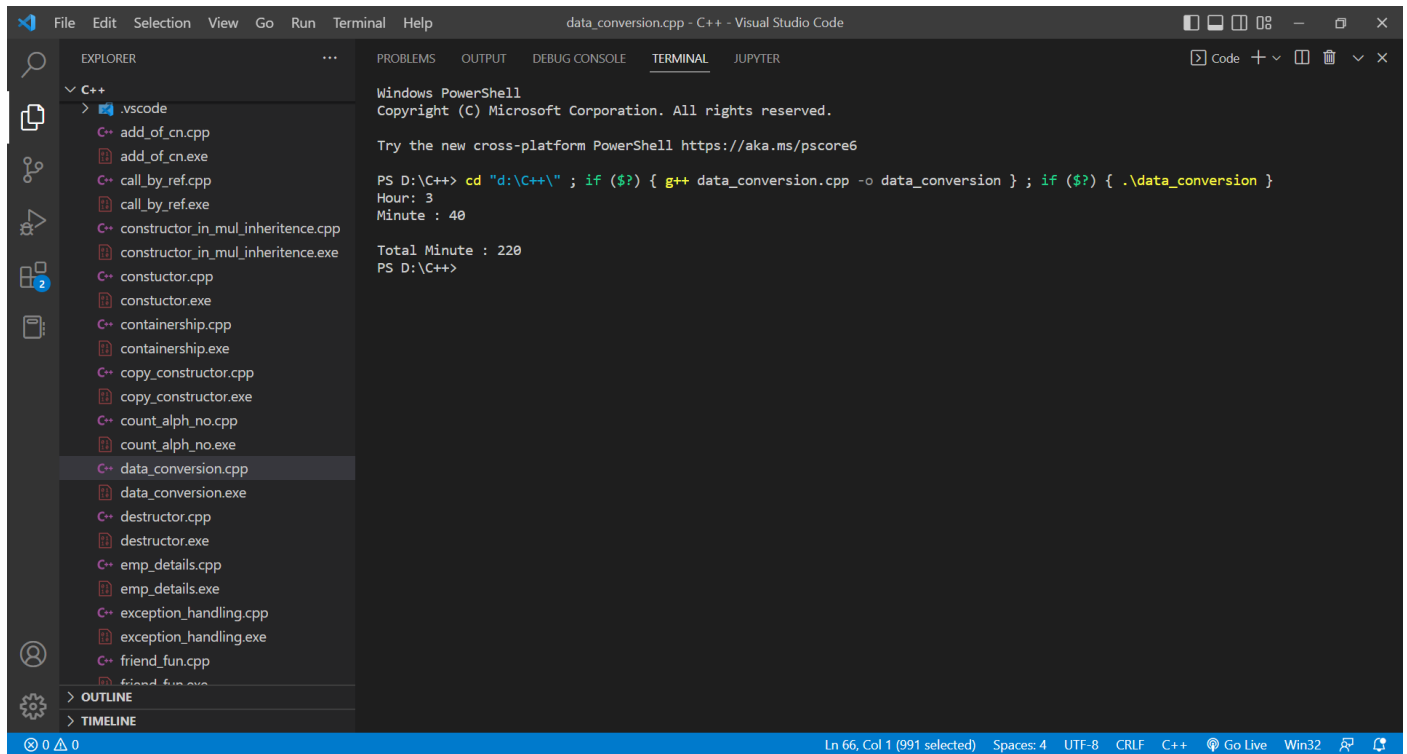
public:
    // Constructors
    Time(int h, int m)
    {
        hr = h;
        mins = m;
    }
    Time()
    {
        cout << "\nTime's Object Created";
    }
    operator Minute () //overloading minute class
    {
        Minute m;
        m.mins = (hr * 60) + mins;
        return m;
    } //driver code

    // Function to print the value of
```

```
// hours and minutes
void show()
{
    cout << "Hour: " << hr << endl;
    cout << "Minute : " << mins << endl;
}
};

// Minutes Class
int main()
{
    Time T1(3,40);
    Minute m;
    m=T1; //minute class is destination and Time class is source class
    T1.show();
    m.show();
    return 0;
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with a C++ project. The Explorer panel on the left lists various source files and executables. The Terminal panel on the right shows the output of a PowerShell command executed in the project directory.

EXPLORER

- ▼ C++
 - > .vscode
 - add_of_cn.cpp
 - add_of_cn.exe
 - call_by_ref.cpp
 - call_by_ref.exe
 - constructor_in_mul_inheritance.cpp
 - constructor_in_mul_inheritance.exe
 - constructor.cpp
 - constructor.exe
 - containership.cpp
 - containership.exe
 - copy_constructor.cpp
 - copy_constructor.exe
 - count_alph_no.cpp
 - count_alph_no.exe
 - data_conversion.cpp
 - data_conversion.exe
 - destructor.cpp
 - destructor.exe
 - emp_details.cpp
 - emp_details.exe
 - exception_handling.cpp
 - exception_handling.exe
 - friend_fun.cpp
 - friend_fun.exe
- > OUTLINE
- > TIMELINE

TERMINAL

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ data_conversion.cpp -o data_conversion } ; if ($?) { .\data_conversion }
Hour: 3
Minute : 40

Total Minute : 220
PS D:\C++>
```

Ln 66, Col 1 (991 selected) Spaces: 4 UTF-8 CRLF C++ Go Live Win32

16.WAP to count digits, alphabets and spaces, stored in a text file, using streams.

CODE: -

```
#include<iostream>
#include<fstream>
using namespace std;

int main()
{

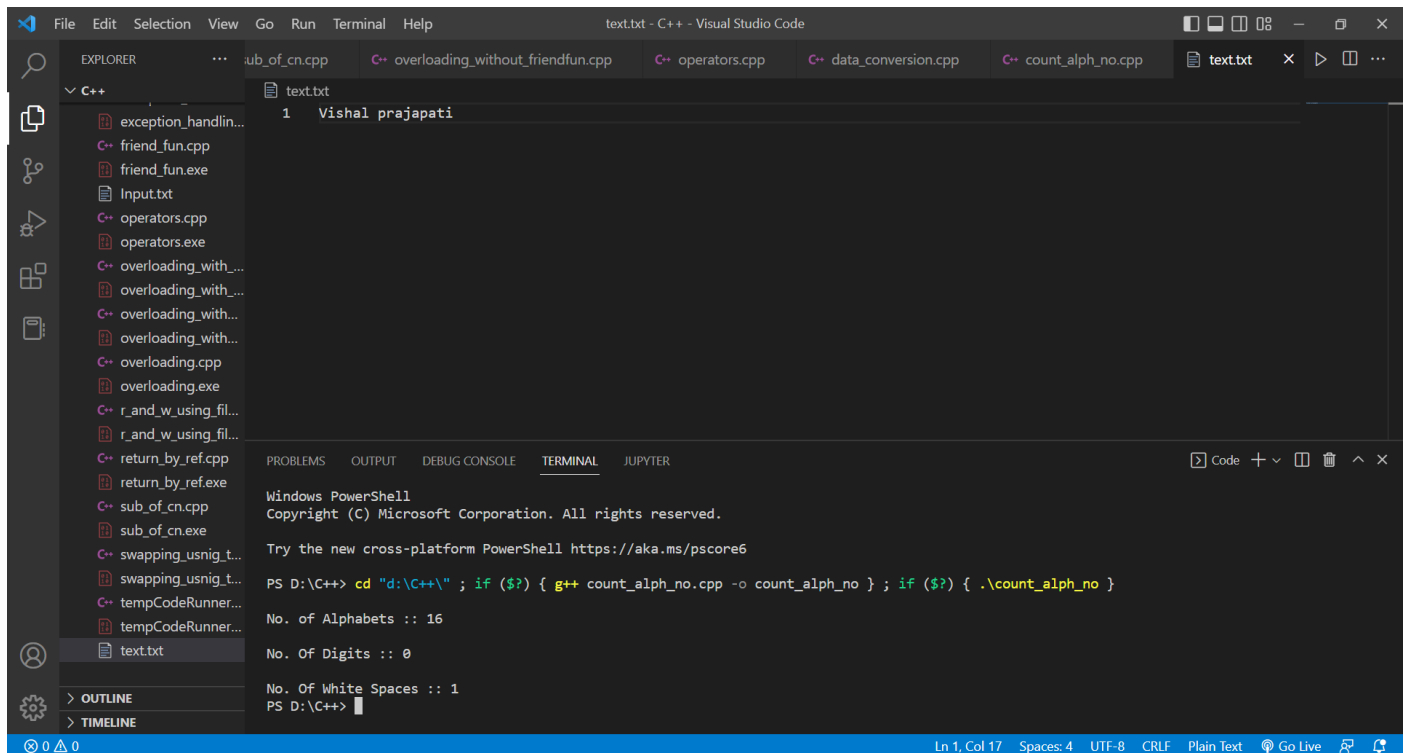
    ifstream fin("text.txt");
    char ch;
    int i,a=0,s=0,d=0;

    while(fin)
    {
        fin.get(ch);
        i=ch;
        if(i>63 && i<91 || i>96 && i<123)
            a++;
        else
            if(ch==' ')
                s++;
            else
                if(i>47&&i<58)
                    d++;
    }

    cout<<"\nNo. of Alphabets :: "<<a<<"\n";
    cout<<"\nNo. Of Digits :: "<<d<<"\n";
    cout<<"\nNo. Of White Spaces :: "<<s<<"\n";

    return 0;
}
```

OUTPUT: -



The screenshot shows the Visual Studio Code interface with a C++ project. The Explorer panel on the left lists files including `exception_handlin...`, `friend_fun.cpp`, `friend_fun.exe`, `Input.txt`, `operators.cpp`, `operators.exe`, `overloading_with_...`, `overloading_with_...`, `overloading_with...`, `overloading_with...`, `overloading.cpp`, `overloading.exe`, `r_and_w_using_fil...`, `r_and_w_using_fil...`, `return_by_ref.cpp`, `return_by_ref.exe`, `sub_of_cn.cpp`, `sub_of_cn.exe`, `swapping_usnig_t...`, `swapping_usnig_t...`, `tempCodeRunner...`, and `tempCodeRunner...`. The main editor displays `text.txt` with the content:

```
1 Vishal prajapati
```

The TERMINAL panel at the bottom shows the output of a PowerShell command:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\C++> cd "d:\C++\" ; if ($?) { g++ count_alph_no.cpp -o count_alph_no } ; if ($?) { .\count_alph_no }

No. of Alphabets :: 16
No. Of Digits :: 0
No. Of White Spaces :: 1
PS D:\C++>
```

The status bar at the bottom indicates the current line and column: `Ln 1, Col 17`, `Spaces: 4`, `UTF-8`, `CRLF`, `Plain Text`, and `Go Live`.