

Assignment-2

Name: P Madeena

Regno: 192211011

1) Create a class Student with name and age. Use a to set name = "Unknown" and age = 0. Display the values.

The screenshot shows the Dev-C++ IDE with a C++ program for a Student class. The code defines a class with attributes name and age, a constructor to initialize them to "Unknown" and 0, and a display method. The main function creates a Student object and calls the display method. The output window shows the program's execution, displaying "Name: Unknown, Age: 0" and then a message indicating the process exited after 0.1497 seconds.

```
practise_blank.cpp
1  #include <iostream>
2  using namespace std;
3
4  class Student {
5      string name;
6      int age;
7  public:
8      Student() {
9          name = "Unknown";
10         age = 0;
11     }
12     void display() {
13         cout << "Name: " << name << ", Age: " << age << endl;
14     }
15 };
16
17 int main() {
18     Student s;
19     s.display();
20     return 0;
21 }
22
```

```
C:\Users\madee\OneDrive\Documents\cpp-c++_programs\practise_blank.cpp - [Executing] - Dev-C++ 5.11
Name: Unknown, Age: 0
-----
Process exited after 0.1497 seconds with return value 0
Press any key to continue . . .
```

Compiler: TDM-GCC 4.9.2 64-bit Release
Errors: 0
Warnings: 0
Output Filename: C:\Users\madee\OneDrive\Documents\cpp-c++_programs\practise_blank.exe
Output Size: 1.8337211608867 MiB
Compilation Time: 0.64s

2) Write a class Rectangle with length and breadth. Initialize them using a and calculate area.

The screenshot shows the Dev-C++ IDE with a C++ program for a Rectangle class. The code defines a class with attributes length and breadth, a constructor to initialize them to 1 and 1, and an area method that calculates the product of length and breadth. The main function creates a Rectangle object with length 5 and breadth 10, and calls the area method. The output window shows the program's execution, displaying "Area: 50" and then a message indicating the process exited after 0.197 seconds.

```
practise_blank.cpp
1  #include <iostream>
2  using namespace std;
3  class Rectangle {
4      int length, breadth;
5  public:
6      Rectangle(int l, int b) {
7          length = l;
8          breadth = b;
9      }
10     int area() {
11         return length * breadth;
12     }
13 };
14 int main() {
15     Rectangle r(5, 10);
16     cout << "Area: " << r.area() << endl;
17     return 0;
18 }
19
```

```
C:\Users\madee\OneDrive\Documents\cpp-c++_programs\practise_blank.cpp - [Executing] - Dev-C++ 5.11
Area: 50
-----
Process exited after 0.197 seconds with return value 0
Press any key to continue . . .
```

Compiler: TDM-GCC 4.9.2 64-bit Release
Errors: 0
Warnings: 0
Output Filename: C:\Users\madee\OneDrive\Documents\cpp-c++_programs\practise_blank.exe
Output Size: 1.83255195617676 MiB
Compilation Time: 0.66s

- 3) Make a class Box with three constructors:
No arguments (set sides = 1)
One argument (cube)
Three arguments (length, width, height).

The screenshot shows the Dev-C++ IDE with a C++ program for a Box class. The code defines three constructors: a default constructor (Box()), a single-argument constructor (Box(int side)), and a three-argument constructor (Box(int l, int w, int h)). A volume() method calculates the volume as length * width * height. The main function creates three Box objects: b1 (default), b2 (side 4), and b3 (length 2, width 3, height 5). It then prints the volume for each object.

```
1 #include <iostream>
2 using namespace std;
3 class Box {
4     int length, width, height;
5 public:
6     Box() {
7         length = width = height = 1;
8     }
9     Box(int side) {
10        length = width = height = side;
11    }
12    Box(int l, int w, int h) {
13        length = l;
14        width = w;
15        height = h;
16    }
17    int volume() {
18        return length * width * height;
19    }
20 };
21 int main() {
22     Box b1, b2(4), b3(2, 3, 5);
23     cout << "Volume1: " << b1.volume() << endl;
24     cout << "Volume2: " << b2.volume() << endl;
25     cout << "Volume3: " << b3.volume() << endl;
26     return 0;
27 }
```

The output window shows the following results:

```
Volume1: 1
Volume2: 64
Volume3: 30

Process exited after 0.1469 seconds with return value 0
Press any key to continue . . .
```

- 4) Define a class Book with title and price. Use a to copy values from one object to another.

The screenshot shows the Dev-C++ IDE with a C++ program for a Book class. The code defines a Book class with a title (string) and price (float). It includes a default constructor, a constructor taking title and price, and a copy constructor (Book(const Book &b)). A display() method prints the title and price. The main function creates a Book object b1 with title "C++ Guide" and price 299.99, then creates a copy b2 = b1 and displays both.

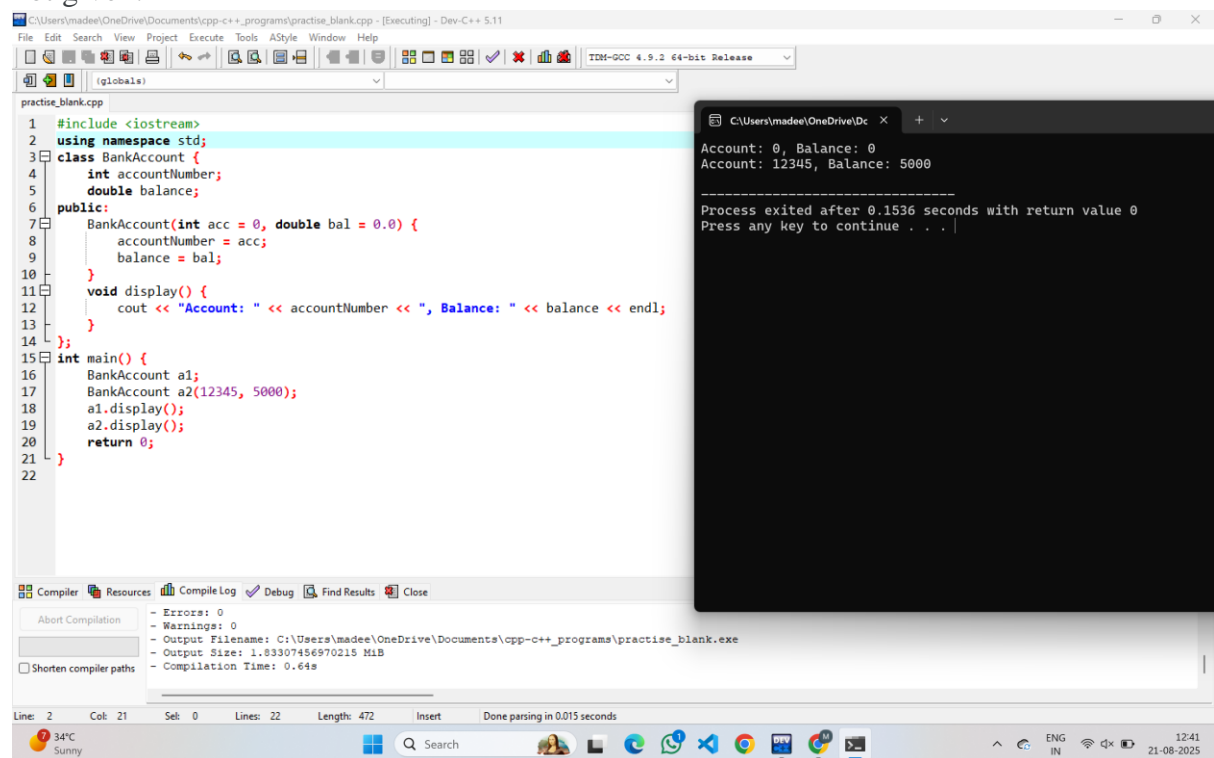
```
1 #include <iostream>
2 using namespace std;
3 class Book {
4     string title;
5     float price;
6 public:
7     Book(string t, float p) {
8         title = t;
9         price = p;
10    }
11    Book(const Book &b) {
12        title = b.title;
13        price = b.price;
14    }
15    void display() {
16        cout << "Title: " << title << ", Price: " << price << endl;
17    }
18 };
19 int main() {
20     Book b1("C++ Guide", 299.99);
21     Book b2 = b1;
22     b1.display();
23     b2.display();
24     return 0;
25 }
```

The output window shows the following results:

```
Title: C++ Guide, Price: 299.99
Title: C++ Guide, Price: 299.99

Process exited after 0.1504 seconds with return value 0
Press any key to continue . . .
```

5) Create a class BankAccount with accountNumber and balance. Use a constructor with if not given.



The screenshot shows the Dev-C++ IDE with a C++ program for a BankAccount class. The code defines a class with two attributes, accountNumber and balance, and a constructor that initializes them. The main function creates two BankAccount objects, one with default values and one with specific values, and calls a display() method to show their details.

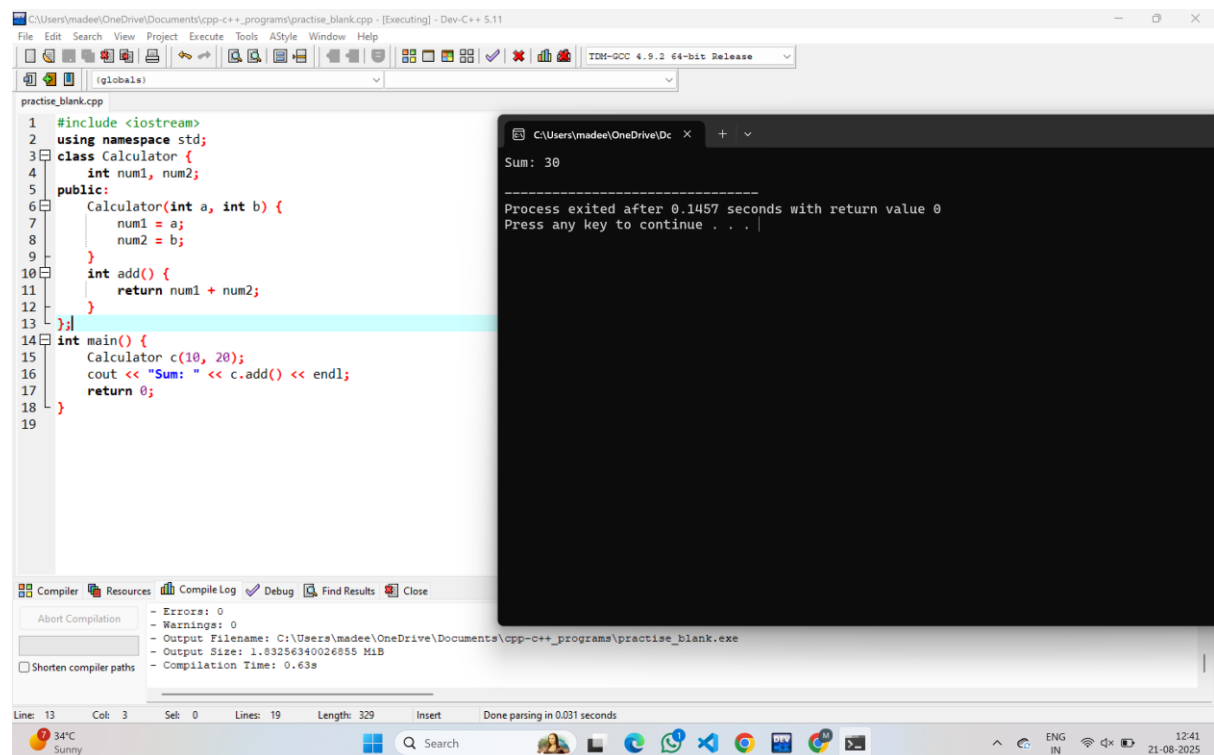
```
1 #include <iostream>
2 using namespace std;
3 class BankAccount {
4     int accountNumber;
5     double balance;
6 public:
7     BankAccount(int acc = 0, double bal = 0.0) {
8         accountNumber = acc;
9         balance = bal;
10    }
11    void display() {
12        cout << "Account: " << accountNumber << ", Balance: " << balance << endl;
13    }
14 };
15 int main() {
16     BankAccount a1;
17     BankAccount a2(12345, 5000);
18     a1.display();
19     a2.display();
20     return 0;
21 }
22
```

The output window shows the execution results:

```
Account: 0, Balance: 0
Account: 12345, Balance: 5000

-----
Process exited after 0.1536 seconds with return value 0
Press any key to continue . . .
```

6) Write a class Calculator with two numbers. Initialize them using a constructor. Add a function to calculate



The screenshot shows the Dev-C++ IDE with a C++ program for a Calculator class. The code defines a class with two attributes, num1 and num2, and a constructor that initializes them. It also includes an add() method to calculate the sum of the two numbers. The main function creates a Calculator object with specific values and calls the add() method to show the result.

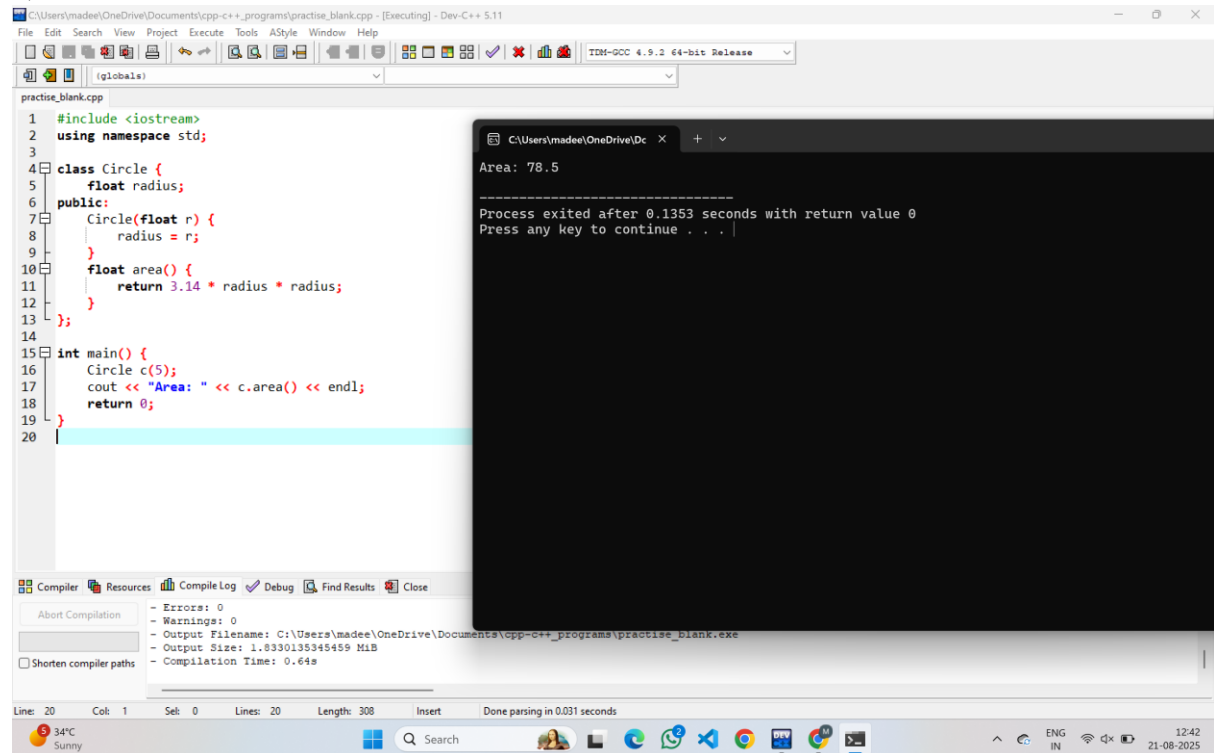
```
1 #include <iostream>
2 using namespace std;
3 class Calculator {
4     int num1, num2;
5 public:
6     Calculator(int a, int b) {
7         num1 = a;
8         num2 = b;
9     }
10    int add() {
11        return num1 + num2;
12    }
13 };
14 int main() {
15     Calculator c(10, 20);
16     cout << "Sum: " << c.add() << endl;
17     return 0;
18 }
19
```

The output window shows the execution results:

```
Sum: 30

-----
Process exited after 0.1457 seconds with return value 0
Press any key to continue . . .
```

7) create a class Circle with radius. Initialize it in the constructor and calculate the area



The screenshot shows a C++ IDE with the following code in `practise_blank.cpp`:

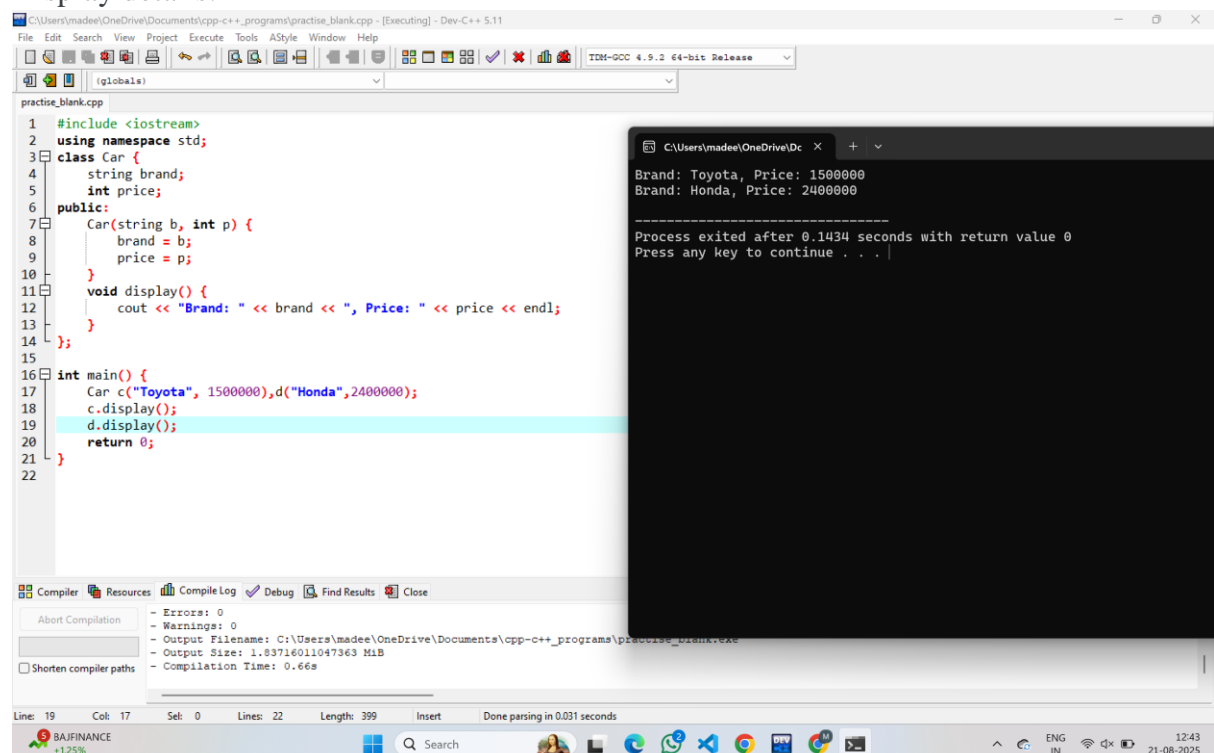
```
1 #include <iostream>
2 using namespace std;
3
4 class Circle {
5     float radius;
6 public:
7     Circle(float r) {
8         radius = r;
9     }
10    float area() {
11        return 3.14 * radius * radius;
12    }
13 };
14
15 int main() {
16     Circle c(5);
17     cout << "Area: " << c.area() << endl;
18     return 0;
19 }
```

The output window shows the following text:

```
Area: 78.5
-----
Process exited after 0.1353 seconds with return value 0
Press any key to continue . . .
```

The compiler output at the bottom shows 0 errors and 0 warnings, with an output file size of 1.8330135345459 MiB and a compilation time of 0.64s.

8) Create a class Car with attributes brand and price. Use a constructor to initialize them. Display details.



The screenshot shows a C++ IDE with the following code in `practise_blank.cpp`:

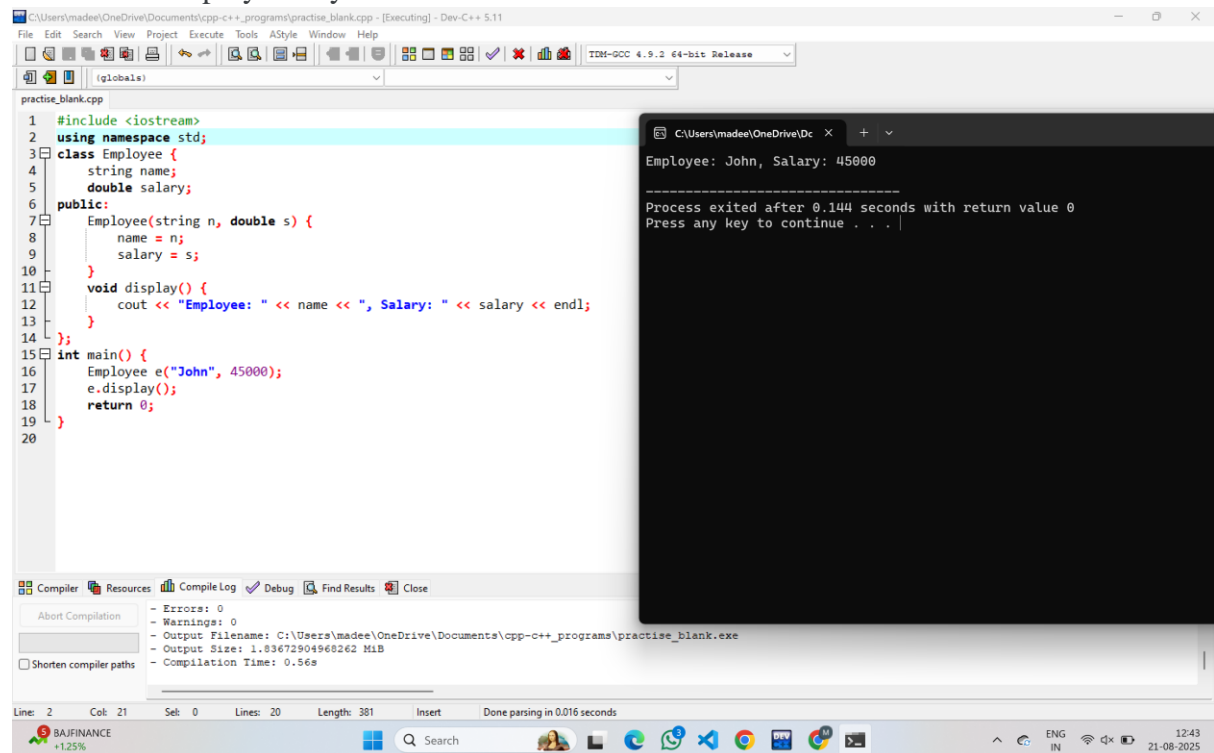
```
1 #include <iostream>
2 using namespace std;
3
4 class Car {
5     string brand;
6     int price;
7 public:
8     Car(string b, int p) {
9         brand = b;
10        price = p;
11    }
12    void display() {
13        cout << "Brand: " << brand << ", Price: " << price << endl;
14    }
15 };
16
17 int main() {
18     Car c("Toyota", 1500000), d("Honda", 2400000);
19     c.display();
20     d.display();
21     return 0;
22 }
```

The output window shows the following text:

```
Brand: Toyota, Price: 1500000
Brand: Honda, Price: 2400000
-----
Process exited after 0.1434 seconds with return value 0
Press any key to continue . . .
```

The compiler output at the bottom shows 0 errors and 0 warnings, with an output file size of 1.83716011047363 MiB and a compilation time of 0.66s.

9) Make a class Employee with name and salary. Use a constructor to set values and a function to display salary details.



The screenshot shows the Dev-C++ IDE with a C++ program for an Employee class. The code defines a class with a constructor and a display function. The main function creates an Employee object and calls the display function. The output window shows the result of the program execution.

```
practise_blank.cpp
1 #include <iostream>
2 using namespace std;
3 class Employee {
4     string name;
5     double salary;
6 public:
7     Employee(string n, double s) {
8         name = n;
9         salary = s;
10    }
11    void display() {
12        cout << "Employee: " << name << ", Salary: " << salary << endl;
13    }
14 };
15 int main() {
16     Employee e("John", 45000);
17     e.display();
18     return 0;
19 }
20
```

Output:

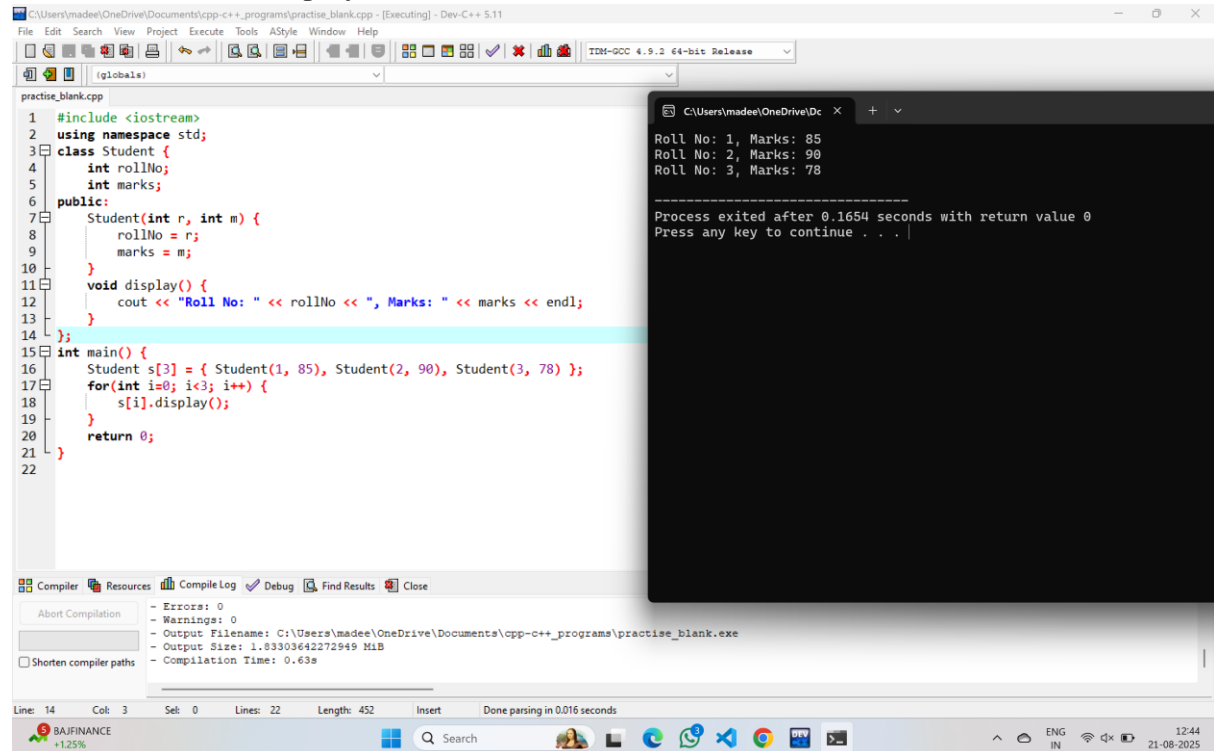
```
Employee: John, Salary: 45000

Process exited after 0.144 seconds with return value 0
Press any key to continue . . .
```

Compiler Output:

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\madee\OneDrive\Documents\cpp-c++_programs\practise_blank.exe
- Output Size: 1.83672904968262 MiB
- Compilation Time: 0.56s
```

10) Create a class Student with rollNo and marks. Initialize each student using a constructor in an array. Display all.



The screenshot shows the Dev-C++ IDE with a C++ program for a Student class. The code defines a class with a constructor and a display function. The main function creates an array of Student objects and calls the display function for each. The output window shows the result of the program execution.

```
practise_blank.cpp
1 #include <iostream>
2 using namespace std;
3 class Student {
4     int rollNo;
5     int marks;
6 public:
7     Student(int r, int m) {
8         rollNo = r;
9         marks = m;
10    }
11    void display() {
12        cout << "Roll No: " << rollNo << ", Marks: " << marks << endl;
13    }
14 };
15 int main() {
16     Student s[3] = { Student(1, 85), Student(2, 90), Student(3, 78) };
17     for(int i=0; i<3; i++) {
18         s[i].display();
19     }
20     return 0;
21 }
22
```

Output:

```
Roll No: 1, Marks: 85
Roll No: 2, Marks: 90
Roll No: 3, Marks: 78

Process exited after 0.1654 seconds with return value 0
Press any key to continue . . .
```

Compiler Output:

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\madee\OneDrive\Documents\cpp-c++_programs\practise_blank.exe
- Output Size: 1.83303642272949 MiB
- Compilation Time: 0.63s
```