

1. Multiplication table

The screenshot shows the Embarcadero Dev-C++ 6.3 IDE. The main window displays the source code for 'Multiplication.cpp'. The code prompts the user to enter a number, reads the input (3), and prints a multiplication table for that number. The console window on the right shows the output: a multiplication table for 3 (3 x 1 = 3 to 3 x 10 = 30) followed by a message indicating the process exited after 1.246 seconds. The compiler window at the bottom shows the compilation details for 'Multiplication.exe'.

```
1 #include <iostream>
2 int main()
3 {
4     int number;
5     std::cout << "Enter a number to print its multiplication table for: ";
6     std::cin >> number;
7     std::cout << "Multiplication table for " << number << ": ";
8     for (int i = 1; i <= 10; ++i) {
9         std::cout << number << " x " << i << " = " << number * i << " ";
10    }
11    return 0;
12 }
```

Output:

```
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30

Process exited after 1.246 seconds with return value 0
Press any key to continue . . .
```

Compiler (2) | Resources | Compile Log | Debug | Find Results | Console | Close

Output Filename: C:\Users\Chennakesi Prudhvi\Videos\Multiplication.exe
Output Size: 2.98873424530029 MiB
Compilation Time: 1.17s

Shorten compiler path

Line: 11 Col: 2 Sel: 0 Lines: 11 Length: 358 Insert Done parsing in 0.015 seconds

2. Strong Number

The screenshot shows the Embarcadero Dev-C++ 6.3 IDE. The main window displays the source code for 'strong number1.cpp'. The code defines a factorial function and an 'isStrongNumber' function. The console window on the right shows the output: 'Enter a number: 0' followed by '0 is a Strong Number.' and a message indicating the process exited after 3.085 seconds. The compiler window at the bottom shows the compilation details for 'strong number1.exe'.

```
1 #include <iostream>
2 using namespace std;
3 int factorial(int n) {
4     int fact = 1;
5     for (int i = 1; i <= n; i++) {
6         fact *= i;
7     }
8     return fact;
9 }
10 bool isStrongNumber(int num) {
11     int originalNum = num;
12     int sum = 0;
13     while (num > 0) {
14         int digit = num % 10;
15         sum += factorial(digit);
16         num /= 10;
17     }
18     return (sum == originalNum);
19 }
20
21 int main() {
22     int n;
23     cout << "Enter a number: ";
24     cin >> n;
25     if (isStrongNumber(n)) {
26         cout << n << " is a Strong Number." << endl;
27     } else {
28         cout << n << " is not a Strong Number." << endl;
29     }
30     return 0;
31 }
```

Output:

```
Enter a number: 0
0 is a Strong Number.

Process exited after 3.085 seconds with return value 0
Press any key to continue . . .
```

Compiler (2) | Resources | Compile Log | Debug | Find Results | Console | Close

Output Filename: C:\Users\Chennakesi Prudhvi\Videos\strong number1.exe
Output Size: 2.98929786682129 MiB
Compilation Time: 1.09s

Shorten compiler path

Line: 8 Col: 17 Sel: 0 Lines: 37 Length: 689 Insert Done parsing in 0.015 seconds

3. Print the pattern

The screenshot shows the Embarcadero Dev-C++ 6.3 IDE. The main window displays a C++ program in `strong number1.cpp` that prints a pattern for `n=5`. The pattern consists of five lines of numbers, where each line `i` contains `i` numbers, and the numbers in each line are the product of the numbers in the previous line. The output window shows the pattern being printed.

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int n = 5;
5     for (int i = 1; i <= n; i++) {
6         for (int j = 1; j <= n - i; j++) {
7             cout << " ";
8         }
9         for (int j = 1; j <= i; j++) {
10            cout << j;
11        }
12        for (int j = i - 1; j >= 1; j--) {
13            cout << j;
14        }
15        cout << endl;
16    }
17    return 0;
18 }
```

Output:

```
1
121
12321
1234321
123454321
```

Process exited after 0.02079 seconds with return value 0
Press any key to continue . . .

4. GCD of two numbers

The screenshot shows the Embarcadero Dev-C++ 6.3 IDE. The main window displays a C++ program in `gcd.cpp` that calculates the GCD of two numbers `n1` and `n2` using the Euclidean algorithm. The program prompts the user to enter the values of `n1` and `n2`. The output window shows the program's execution with the input values 2 and 4, and the resulting GCD of 2.

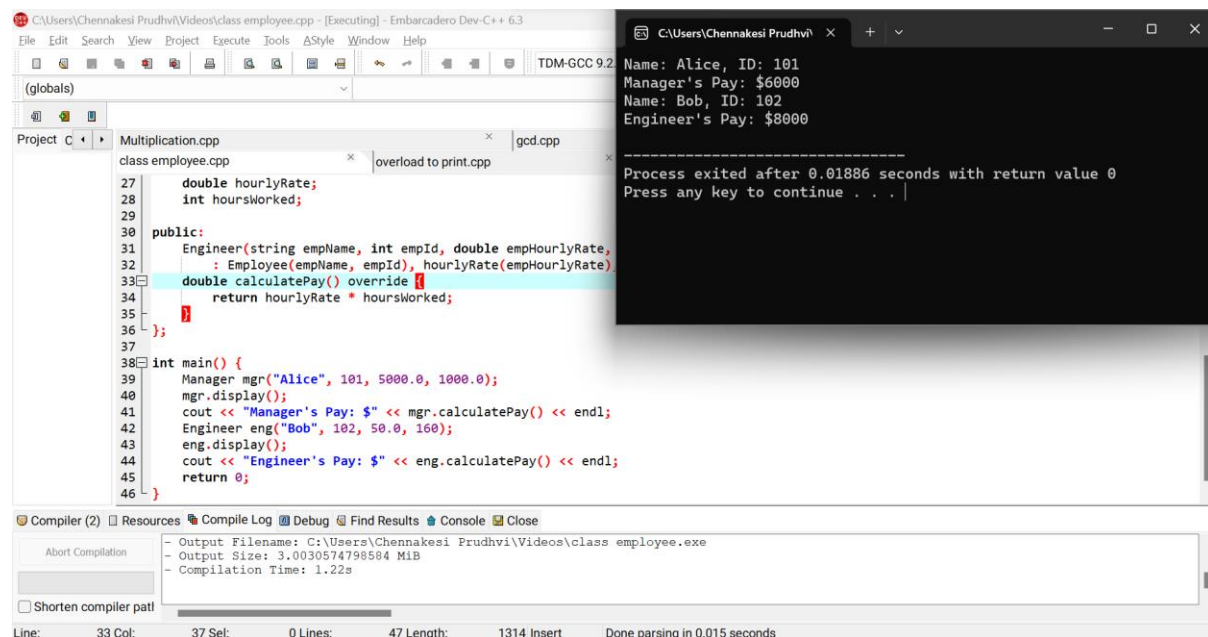
```
2 using namespace std;
3 int gcd(int n1, int n2)
4 {
5     int i, gcd;
6     for (i = 1; i <= n1 && n2; i++)
7     {
8         if (n1 % i == 0 && n2 % i == 0)
9         {
10            gcd = i;
11        }
12    }
13    return gcd;
14 }
15 int main()
16 {
17     int n1, n2;
18     cout << "enter the values of n1 and n2:";
19     cin >> n1 >> n2;
20     cout << gcd(n1, n2);
21 }
22 }
```

Output:

```
enter the values of n1 and n2: 2 4
2
```

Process exited after 2.301 seconds with return value 0
Press any key to continue . . .

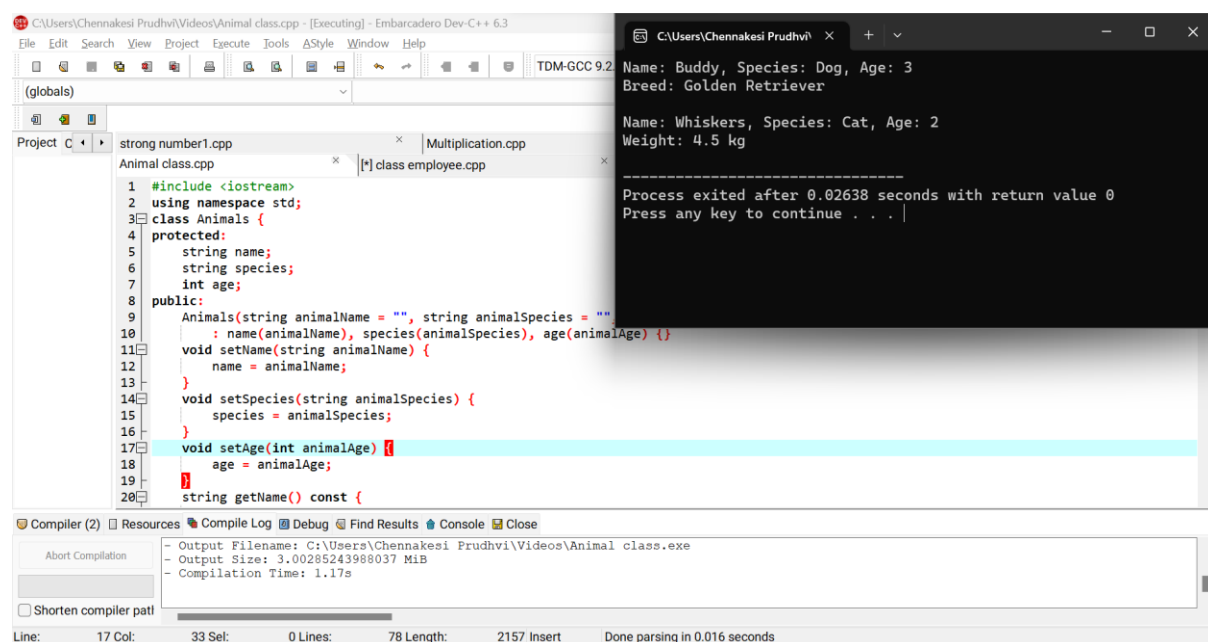
5. Base class for Employee and Manager



```
C:\Users\Chennakesi Prudhvi\Videos\class employee.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project C:\ Multiplication.cpp class employee.cpp overload to print.cpp gcd.cpp
27 double hourlyRate;
28 int hoursWorked;
29
30 public:
31     Engineer(string empName, int empId, double empHourlyRate,
32               : Employee(empName, empId), hourlyRate(empHourlyRate))
33     double calculatePay() override {
34         return hourlyRate * hoursWorked;
35     }
36 };
37
38 int main() {
39     Manager mgr("Alice", 101, 5000.0, 1000.0);
40     mgr.display();
41     cout << "Manager's Pay: $" << mgr.calculatePay() << endl;
42     Engineer eng("Bob", 102, 50.0, 160);
43     eng.display();
44     cout << "Engineer's Pay: $" << eng.calculatePay() << endl;
45     return 0;
46 }
Compiler (2) Resources Compile Log Debug Find Results Console Close
Abort Compilation
- Output Filename: C:\Users\Chennakesi Prudhvi\Videos\class employee.exe
- Output Size: 3.0030574798584 MiB
- Compilation Time: 1.22s
Shorten compiler path
Line: 33 Col: 37 Sel: 0 Lines: 47 Length: 1314 Insert Done parsing in 0.015 seconds
```

```
C:\Users\Chennakesi Prudhvi\
Name: Alice, ID: 101
Manager's Pay: $6000
Name: Bob, ID: 102
Engineer's Pay: $8000
-----
Process exited after 0.01886 seconds with return value 0
Press any key to continue . . .
```

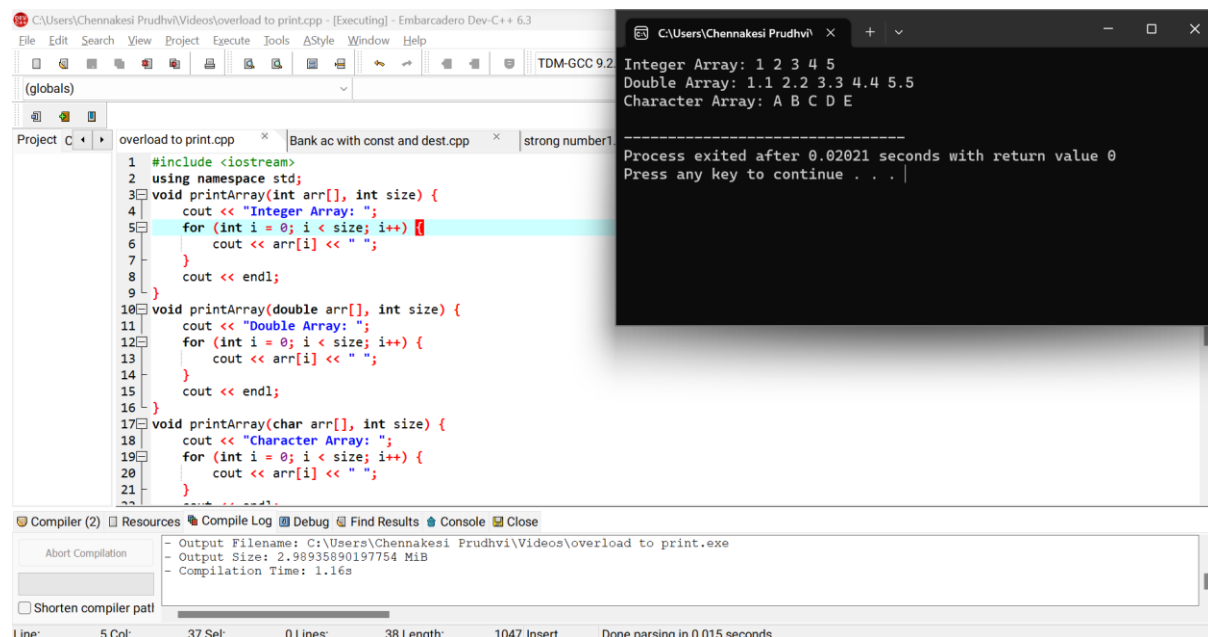
6. Base class animal



```
C:\Users\Chennakesi Prudhvi\Videos\Animal class.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project C:\ strong number1.cpp Multiplication.cpp
Animal class.cpp class employee.cpp
1 #include <iostream>
2 using namespace std;
3 class Animals {
4     protected:
5         string name;
6         string species;
7         int age;
8     public:
9         Animals(string animalName = "", string animalSpecies = "",
10                 : name(animalName), species(animalSpecies), age(animalAge) {}
11         void setName(string animalName) {
12             name = animalName;
13         }
14         void setSpecies(string animalSpecies) {
15             species = animalSpecies;
16         }
17         void setAge(int animalAge) {
18             age = animalAge;
19         }
20         string getName() const {
21             return name;
22         }
23     };
24 }
25
26 int main() {
27     Animals a("Buddy", "Dog", 3);
28     a.setName("Buddy");
29     a.setSpecies("Golden Retriever");
30     a.setAge(3);
31     a.display();
32     Animals b("Whiskers", "Cat", 2);
33     b.setName("Whiskers");
34     b.setSpecies("Cat");
35     b.setAge(2);
36     b.display();
37     return 0;
38 }
Compiler (2) Resources Compile Log Debug Find Results Console Close
Abort Compilation
- Output Filename: C:\Users\Chennakesi Prudhvi\Videos\Animal class.exe
- Output Size: 3.00285243988037 MiB
- Compilation Time: 1.17s
Shorten compiler path
Line: 17 Col: 33 Sel: 0 Lines: 78 Length: 2157 Insert Done parsing in 0.016 seconds
```

```
C:\Users\Chennakesi Prudhvi\
Name: Buddy, Species: Dog, Age: 3
Breed: Golden Retriever
Name: Whiskers, Species: Cat, Age: 2
Weight: 4.5 kg
-----
Process exited after 0.02638 seconds with return value 0
Press any key to continue . . .
```

7. Print the contents of integer, double and character array



The screenshot shows the Embarcadero Dev-C++ 6.3 IDE. The main window displays a C++ program named 'overload to print.cpp'. The program defines three functions: `printArray` for integer arrays, `printArray` for double arrays, and `printArray` for character arrays. The `main` function calls these three functions with appropriate data. The console window on the right shows the output of the program, which prints the contents of the arrays. The compiler window at the bottom shows the compilation details.

```
1 #include <iostream>
2 using namespace std;
3 void printArray(int arr[], int size) {
4     cout << "Integer Array: ";
5     for (int i = 0; i < size; i++) {
6         cout << arr[i] << " ";
7     }
8     cout << endl;
9 }
10 void printArray(double arr[], int size) {
11     cout << "Double Array: ";
12     for (int i = 0; i < size; i++) {
13         cout << arr[i] << " ";
14     }
15     cout << endl;
16 }
17 void printArray(char arr[], int size) {
18     cout << "Character Array: ";
19     for (int i = 0; i < size; i++) {
20         cout << arr[i] << " ";
21     }
22     cout << endl;
23 }
```

Integer Array: 1 2 3 4 5
Double Array: 1.1 2.2 3.3 4.4 5.5
Character Array: A B C D E

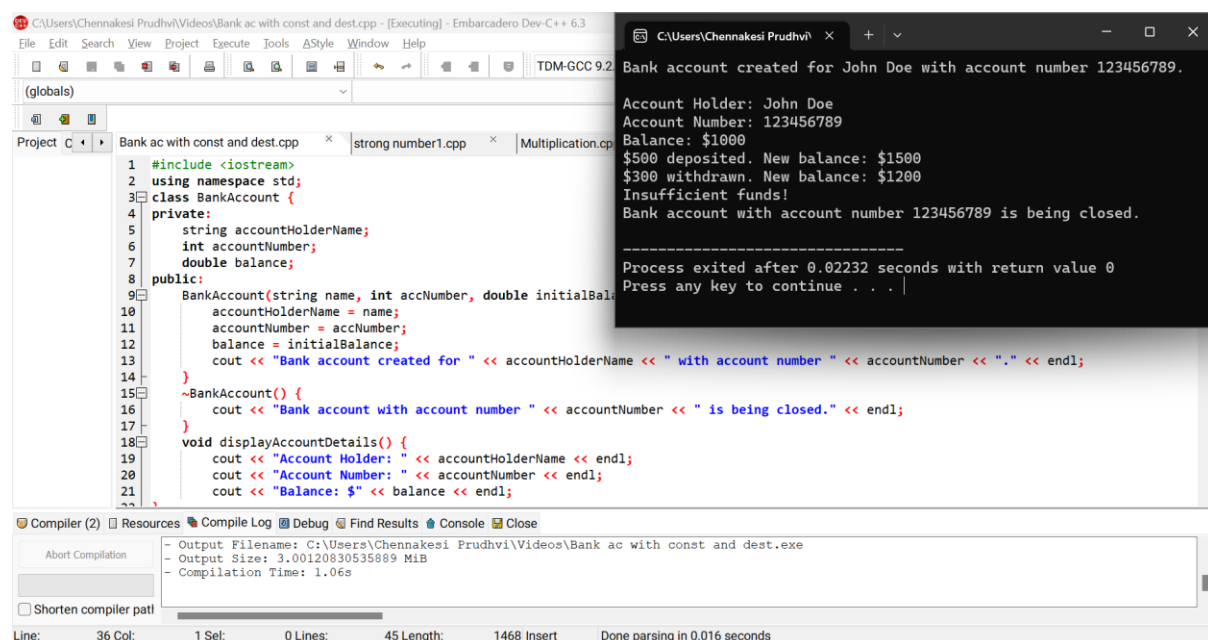
Process exited after 0.02021 seconds with return value 0
Press any key to continue . . .

Compiler (2) | Resources | Compile Log | Debug | Find Results | Console | Close

Output Filename: C:\Users\Chennakesi Prudhvi\Videos\overload to print.exe
Output Size: 2.98935890197754 MiB
Compilation Time: 1.16s

Line: 5 Col: 37 Sel: 0 Lines: 38 Length: 1047 Insert Done parsing in 0.015 seconds

8. Bank account for constructor and destructor



The screenshot shows the Embarcadero Dev-C++ 6.3 IDE. The main window displays a C++ program named 'Bank ac with const and dest.cpp'. The program defines a `BankAccount` class with a constructor, a destructor, and a `displayAccountDetails` method. The `main` function creates a `BankAccount` object, displays its details, and then destroys it. The console window on the right shows the output of the program, which prints the account details and the messages from the constructor and destructor. The compiler window at the bottom shows the compilation details.

```
1 #include <iostream>
2 using namespace std;
3 class BankAccount {
4 private:
5     string accountHolderName;
6     int accountNumber;
7     double balance;
8 public:
9     BankAccount(string name, int accNumber, double initialBal) {
10         accountHolderName = name;
11         accountNumber = accNumber;
12         balance = initialBalance;
13         cout << "Bank account created for " << accountHolderName << " with account number " << accountNumber << "." << endl;
14     }
15     ~BankAccount() {
16         cout << "Bank account with account number " << accountNumber << " is being closed." << endl;
17     }
18     void displayAccountDetails() {
19         cout << "Account Holder: " << accountHolderName << endl;
20         cout << "Account Number: " << accountNumber << endl;
21         cout << "Balance: $" << balance << endl;
22     }
23 }
```

Bank account created for John Doe with account number 123456789.

Account Holder: John Doe
Account Number: 123456789
Balance: \$1000
\$500 deposited. New balance: \$1500
\$300 withdrawn. New balance: \$1200
Insufficient funds!
Bank account with account number 123456789 is being closed.

Process exited after 0.02232 seconds with return value 0
Press any key to continue . . .

Compiler (2) | Resources | Compile Log | Debug | Find Results | Console | Close

Output Filename: C:\Users\Chennakesi Prudhvi\Videos\Bank ac with const and dest.exe
Output Size: 3.00120830535889 MiB
Compilation Time: 1.06s

Line: 36 Col: 1 Sel: 0 Lines: 45 Length: 1468 Insert Done parsing in 0.016 seconds