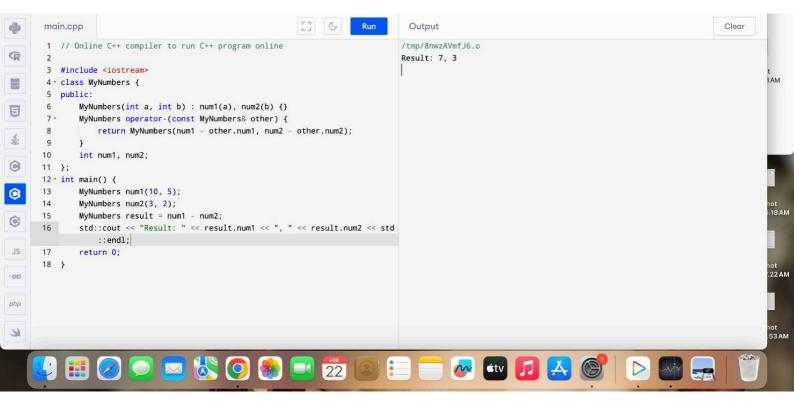
```
[] G Run
      main.cpp
                                                                          Output
                                                                                                                                    Clear
4
      1 //3.overload the << operator to print contents of a user defined class
                                                                         /tmp/HqOWG3SWV2.o
R
                                                                         a: 10, b: 20
      3 #include <iostream>
5 - class MyClass {
      6
            int a, b;
9
      7 public:
            MyClass(int x, int y) : a(x), b(y) { }
鱼
      9
            friend std::ostream& operator<<(std::ostream& os, const MyClass& obj);</pre>
      10
0
      11 };
      G
      15
            return os;
0
      16 }
      17
JS
      18 - int main() {
            MyClass myObj(10, 20);
      19
      20
            std::cout << myObj << std::endl;</pre>
      21
      22
            return 0;
php
      23 }
(B)
```

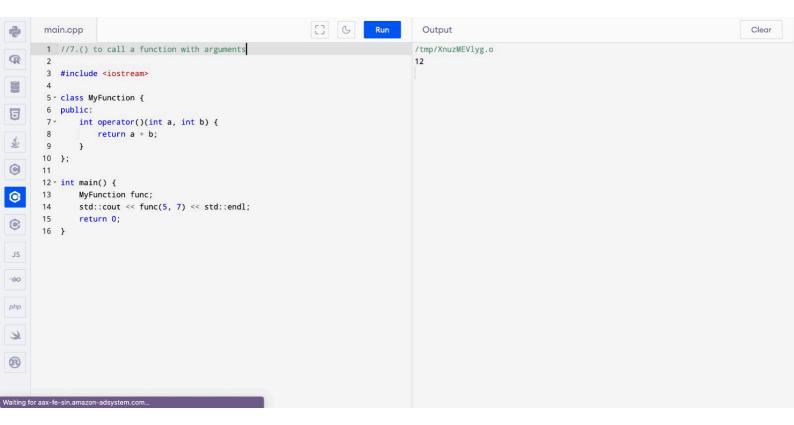
```
[] 6
       main.cpp
                                                                                   Output
                                                                                                                                                     Clear
4
       1 //1.overload the + operator to add two variables
                                                                                  /tmp/7osvAkHbWb.o
R
                                                                                  8
       3 #include <iostream>
5 - class MyClass {
       6
             int var;
5
       7 public:
             MyClass(int x = 0) : var(x) { }
鱼
       9
             MyClass operator+(const MyClass& other) const {
       10 -
0
       11
                return MyClass(var + other.var);
       12
G
       13
              friend std::ostream& operator<<(std::ostream& os, const MyClass& obj);</pre>
       14
       15 };
0
       17 * std::ostream& operator<<(std::ostream& os, const MyClass& obj) {
JS
           os << obj.var;
       18
       19
              return os;
       20 }
      21
      22 - int main() {
php
            MyClass a(5), b(3);
      23
       24
              MyClass c = a + b;
      25
             std::cout << c << std::endl; // Output: 8
      26
8
      27
              return 0;
      28 }
      29
```

```
[] 6
       main.cpp
                                                                                    Output
                                                                                                                                                      Clear
4
       1 //1.overload the ++ operator to increment a variable
                                                                                  /tmp/7osvAkHbWb.o
R
                                                                                  5
       3 #include <iostream>
                                                                                  5
4 - class MyClass {
             int var;
       6 public:
9
              MyClass(int x = 0) : var(x) { }
雪
       9 +
              MyClass operator++() {
                 var++;
       10
0
       11
                  return *this;
       12
•
       13
              friend std::ostream& operator<<(std::ostream& os, const MyClass& obj);</pre>
       14 };
       15 * std::ostream& operator<<(std::ostream& os, const MyClass& obj) {
0
       16
             os << obj.var;
       17
              return os;
JS
       18 }
       19
       20 * int main() {
            MyClass a(5);
      21
      22
              std::cout << a << std::endl;
php
             'a++';
       23
      24
            std::cout << a << std::endl;
      25
      26
              return 0;
(B)
      27 }
      28
```

```
[] G Run
       main.cpp
                                                                                       Output
                                                                                                                                                           Clear
4
       1 //4.overload the == operator to compare two objects of a user defined class /tmp/HqOWG3SWV2.o
R
                                                                                     a and b are equal
       3 #include <iostream>
                                                                                     a and c are not equal
4 - class MyClass {
              int x, y;
       6 public:
9
              MyClass(int a, int b) : x(a), y(b) { }
1
       9 +
              bool operator==(const MyClass& other) const {
                 return (x == other.x && y == other.y);
       10
0
       11
       12 };
(3
       13 * int main() {
              MyClass a(5, 6), b(5, 6), c(7, 8);
       14
              if (a == b)
       15
0
       16
                  std::cout << "a and b are equal" << std::endl;
       17
              else
JS
                 std::cout << "a and b are not equal" << std::endl;</pre>
       18
              if (a == c)
       19
       20
                  std::cout << "a and c are equal" << std::endl;</pre>
       21
                  std::cout << "a and c are not equal" << std::endl;</pre>
       22
php
       23
               return 0;
       24 }
(B)
```



```
[] G Run
       main.cpp
                                                                                     Output
                                                                                                                                                        Clear
4
       1 //6. [] operator
                                                                                   /tmp/XnuzMEVlyg.o
R
                                                                                   20
       3 #include <iostream>
5 - class MyArray {
       6 private:
9
              int* data;
             int size;
雪
       9
       10 public:
0
       11 -
             MyArray(int size) : size(size) {
       12
                 data = new int[size];
©
       13
       14
              int& operator[](int index) {
       15 +
0
       16 +
               if (index < 0 || index >= size) {
                      std::cout << "Index out of bounds!" << std::endl;
       17
JS
                      exit(1);
       18
       19
       20
                 return data[index];
       21
       22
php
              // Other methods and destructor
       23
       24 };
       25
       26 - int main() {
(B)
           MyArray arr(5);
arr[0] = 10;
arr[1] = 20;
       27
       28
       29
            ctd..cout << arr[1] << ctd..ond].
```



```
[] 6
       main.cpp
                                                                                    Output
                                                                                                                                                       Clear
4
       1 //9.overload a function to add two integer numbers and two floating point
                                                                                   /tmp/XnuzMEVlyg.o
R
            number separately
                                                                                   12
       2
                                                                                   13.2
3 #include <iostream>
       5 - class MyMath {
5
       6 public:
             int addInt(int a, int b) {
1
       8
                 return a + b;
       9
0
       10
       11 *
              double addFloat(double a, double b) {
•
       12
                 return a + b;
       13
       14 };
0
       15
       16 * int main() {
JS
            MyMath math;
       17
       18
              std::cout << math.addInt(5, 7) << std::endl;</pre>
              std::cout << math.addFloat(5.5, 7.7) << std::endl;</pre>
       19
      20
             return 0;
      21 }
php
R
(B)
```