**Quiz 4**

1. **There are three general cases where the copy constructor is called instead of the assignment operator?**

|  |  |
| --- | --- |
| **A** | When instantiating one object and initializing it with values from another object |
| **B** | When passing an object by value |
| **C** | When an object is returned from a function by value |
| **D** | All the above |

1. **Which of the following operators cannot be overloaded**

|  |  |
| --- | --- |
| **A** | . (Member Access or Dot operator) |
| **B** | ?: (Ternary or Conditional Operator ) |
| **C** | :: (Scope Resolution Operator) |
| **D** | .\* (Pointer-to-member Operator ) |
| **E** | All of the above |

1. **Which operator does the job similar to** **Copy Constructor?**

|  |  |
| --- | --- |
| **A** | == |
| **B** | = |
| **C** | ++ |
| **D** | + |

1. **A reason to overload the \_\_\_\_\_\_\_\_\_\_\_\_  is to write classes that have array-like behaviors.**

|  |  |
| --- | --- |
| **A** | parentheses  ( ) operator |
| **B** | curly braces  { } operator |
| **C** | square brackets [ ] operator |
| **D** | colon : : operator |

1. **How does C++ compiler differs between overloaded postfix and prefix operators?**

|  |  |
| --- | --- |
| **A** | C++ doesn't allow both operators to be overloaded in a class |
| **B** | A postfix ++ has a dummy parameter |
| **C** | A prefix ++ has a dummy parameter |
| **D** | By making prefix ++ as a global function and postfix as a member function. |

1. **Predict the output**

*#include<iostream>*

*using namespace std;*

*class A*

*{*

*int i;*

*public:*

*A(int ii = 0) : i(ii) {}*

*void show() {  cout << i << endl;  }*

*};*

*class B*

*{*

*int x;*

*public:*

*B(int xx) : x(xx) {}*

*operator A() const {  return A(x); }*

*};*

*void g(A a)*

*{*

*a.show();*

*}*

*int main()*

*{*

*B b(10);*

*g(b);*

*g(20);*

*return 0;*

*}*

**Output:**

|  |  |
| --- | --- |
| **A** | Compiler Error |
| **B** | 10  20 |
| **C** | 20  20 |
| **D** | 10  10 |

1. **Output of following program?**

*#include <iostream>*

*using namespace std;*

*class Test2*

*{*

*int y;*

*};*

*class Test*

*{*

*int x;*

*Test2 t2;*

*public:*

*operator Test2 ()  { return t2; }*

*operator int () { return x; }*

*};*

*void fun ( int x) { cout << "fun(int) called"; }*

*void fun ( Test2 t ) { cout << "fun(Test 2) called"; }*

*int main()*

*{*

*Test t;*

*fun(t);*

*return 0;*

*}*

|  |  |
| --- | --- |
| **A** | fun(int) called |
| **B** | fun(Test 2) called |
| **C** | Compiler Error: Ambiguous call to fun() |

1. **Predict the output?**

|  |
| --- |
| *#include<stdlib.h>*  *#include<stdio.h>*  *#include<iostream>*    *using namespace std;*    *class Test {*  *int x;*  *public:*  *void\* operator new(size\_t size);*  *void operator delete(void\*);*  *Test(int i) {*  *x = i;*  *cout << "Constructor called \n";*  *}*  *~Test() { cout << "Destructor called \n"; }*  *};*      *void\* Test::operator new(size\_t size)*  *{*  *void \*storage = malloc(size);*  *cout << "new called \n";*  *return storage;*  *}*    *void Test::operator delete(void \*p )*  *{*  *cout<<"delete called \n";*  *free(p);*  *}*    *int main()*  *{*  *Test \*m = new Test(5);*  *delete m;*  *return 0;*  *}* |

**Output:**

|  |  |
| --- | --- |
| **A** | new called  Constructor called  delete called  Destructor called |
| **B** | new called  Constructor called  Destructor called  delete called |
| **C** | Constructor called  new called  Destructor called  delete called |
| **D** | Constructor called  new called  delete called  Destructor called |



|  |  |
| --- | --- |
|  | *#include<iostream>*  *using namespace std;*    *class Point {*  *private:*  *int x, y;*  *public:*  *Point() : x(0), y(0) { }*  *Point& operator()(int dx, int dy);*  *void show() {cout << "x = " << x << ", y = " << y; }*  *};*    *Point& Point::operator()(int dx, int dy)*  *{*  *x = dx;*  *y = dy;*  *return \*this;*  *}*    *int main()*  *{*  *Point pt;*  *pt(3, 2);*  *pt.show();*  *return 0;*  *}* |

**Output:**

|  |  |
| --- | --- |
| **A** | x = 3, y = 2 |
| **B** | Compiler Error |
| **C** | x = 2, y = 3 |

1. **Which of the following operator functions cannot be global, i.e., must be a member function.**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **A** | new |
| **B** | delete |
| **C** | Conversation Operator |
| **D** | All of the above |