Unit 5 - Week 3

Assignment 3

The due date for submitting this assignment has passed. Due on 2020-02-19, 23:59 IST. As per our records you have not submitted this assignment.

1) Consider the code segment below.

```cpp
class myClass {
    // code...
};
```

```cpp
int main() {
    const myClass ob;    // LINE-1
    return 0;
}
```

What is the type of this pointer associated with the object `ob`?

- a) `const myClass* this;`
- b) `myClass const* this;`
- c) `myClass* const this;`
- d) `myClass const* const this;`

No, the answer is incorrect.

Score: 0

Accepted Answers:
d) `myClass const* const this;`

2) Consider the program below.

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

typedef struct Complex {
    mutable double re;
    mutable double im;
} Complex;

int main() {
    const Complex c = { 2, 4 };
    c.re = 5;
    cout << c.re << " , " << c.im;
    return 0;
}
```

What will be the output / error?

- a) 2, 4
- b) 5, 4
- c) Compilation error: constant object c cannot be assign to {2, 4}
- d) Compilation error: Complex::re is a read-only object

No, the answer is incorrect.
Score: 0
Accepted Answers:

- b) 5, 4
3) Consider the program below.

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

class Point {
    // data-members
    public:
        Point() { cout << "c1 "; }
        Point(const Point &t) { cout << "c2 "; }
        Point& operator=(const Point& c) {
            // code to assign the data-members
            cout << "c3 ";
            return *this;
        }
    }

    int main(){
        Point *t1, *t2;  //LINE-1
        t1 = new Point();  //LINE-2
        t2 = new Point(*t1);  //LINE-3
        Point t3 = *t1;  //LINE-4
        Point t4;  //LINE-5
        t4 = t3;  //LINE-6
        return 0;
    }
}
```

What will be the output of the above code?

- a) c1 c1 c2 c2 c1 c3
- b) c2 c2 c3 c1 c3
- c) c1 c2 c3 c1 c3
- d) c1 c2 c2 c1 c3

No, the answer is incorrect.
Score: 0
Accepted Answers:
```
d) c1 c2 c2 c1 c3
```
4) Consider the program below.

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

int set_a(int i) {
    cout << "init a : " << i << endl;
    return i;
}

int set_b(int i) {
    cout << "init b : " << i << endl;
    return i;
}

int set_c(int i) {
    cout << "init c : " << i << endl;
    return i;
}

class MyClass {
    int ___________; // LINE-1: declare the data members
public:
    MyClass(int _a, int _b, int _c) : a(set_a(_a)),
        b(set_b(_b)), c(set_c(_c)){}
};

int main() {
    MyClass obj(1, 2, 3);
    return 0;
}
```

Fill in the blank at LINE-1 such that the output is as follows:

init c : 3
init b : 2
init a : 1

- a) _a, _b, _c
- b) _c, _b, _a
- c) a, b, c
- d) c, b, a

No, the answer is incorrect.
Score: 0
Accepted Answers:
- d) c, b, a

5)
Consider the program below.

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

class Point {
  int x, y;
public:
  Point(int _x, int _y) : x(_x), y(_y) {}
  Point(Point& pt) { this = pt; }
  void show() { cout << x << " ", " << y << endl; }
};

int main() {
  Point p(10, 20);
  Point p1(p);

  p1.show();

  return 0;
}
```

What will be output / error?

- a) 0, 0
- b) 10, 20
- c) Compiler Error: lvalue required as left operand of assignment
- d) Compiler Error: private x, y are inaccessible

No, the answer is incorrect.
Score: 0
Accepted Answers:
- c) Compiler Error: lvalue required as left operand of assignment

6) 2 points
Consider the program below.

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

class myClass {
public:
    bool isSame(myClass&);
};

bool myClass::isSame(myClass& ob) {
    if (&ob == this)
        return true;
    else
        return false;
}

int main() {
    myClass ob1;
    myClass *ob2 = &ob1;    // LINE-1
    cout << ob2->isSame(ob1);
    return 0;
}
```

What will be the output?

- a) 0
- b) 1
- c) false
- d) true

No, the answer is incorrect.
Score: 0
Accepted Answers:

b) 1

2 points

https://onlinecourses.nptel.ac.in/noc20_cs07/unit/unit=5&assessment=109
Consider the program below.

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

class MyClass {
public:
    MyClass() { }  // default constructor
    MyClass(const MyClass& c2) { }  // copy constructor
    MyClass& func(MyClass &obj2) {
        MyClass &obj3 = obj2;
        return obj3;
    }
};

int main() {
    MyClass obj1;
    obj1.func(obj1);
    return 0;
}
```

How many times copy constructor will be invoked?

- a) 0
- b) 1
- c) 2
- d) 3

No, the answer is incorrect.
Score: 0
Accepted Answers:
- a) 0
8) Consider the program segment.

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

int i = 0;

class A {
  public:
    ~A() { i = 10; }
};

int fun() {
  i = 3;

  A obj; }

  return i++;
}

int main() {
  cout << fun() << " ";
  cout << i << endl;

  return 0;
}

What will be the output?

- a) 3 10
- b) 3 4
- c) 10 11
- d) 4 10

No, the answer is incorrect.
Score: 0
Accepted Answers:
- c) 10 11

9)

```
Consider the following program.

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

class Complex {
  int re, im;
public:
  Complex(int _re = 0, int _im = 0) : re(_re), im(_im) {}
  int get_re() const { return re; }
  int get_im() { return im; }
};

int main() {
  const Complex c;

  cout << c.get_re() << " ";  // Line-1
  cout << c.get_im();        // Line-2

  return 0;
}
```

What will be the output?

- a) 0 0
- b) garbage values
- c) Compilation Error: at Line-1
- d) Compilation Error: at Line-2

No, the answer is incorrect.
Score: 0
Accepted Answers:

d) Compilation Error: at Line-2
10)

What will be the output of the following program?

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

class Sample {
    string name;
public:
    Sample(string s) : name(s) {
        cout << name << " Created" << " ";
    }
    ~Sample() {
        cout << name << " Destroyed" << " ";
    }
};

int main() {
    Sample s1("S1"), s2("S2");
    return 0;
}
```

- a) S1 Created S2 Created S1 Destroyed S2 Destroyed
- b) S1 Created S2 Created S2 Destroyed S1 Destroyed
- c) S2 Created S1 Created S2 Destroyed S1 Destroyed
- d) S1 Created S1 Destroyed S2 Created S2 Destroyed

No, the answer is incorrect.
Score: 0
Accepted Answers:
- b) S1 Created S2 Created S2 Destroyed S1 Destroyed