Unit 6 - Week 4

Assignment 4

The due date for submitting this assignment has passed. **Due on 2020-02-26, 23:59 IST.**
As per our records you have not submitted this assignment.

1) **2 points**
Consider the following program.

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

class B {
    int _b;
public:
    B() : _b(1) { }
    int func(A& objA);
};

class A {
    int _a;          // LINE-1
public:
    A() { _a = 0; }
    friend int B::func(A&); // LINE-2
};

int B::func(A &objA) {
    return (objA._a + _b); // LINE-3
}

int main() {
    A oA;
    B oB;
    cout << "Result:" << oB.func(oA);
    return 0;
}
```

What will be the output/error of the above code:

a) Result: 1

b) Compiler error as int A::A is private but used in function B::func() at LINE-3

c) Compilation error: Syntax Error: Identifier A

d) Result: 0

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

c) Compilation error: Syntax Error: Identifier A

2)
Consider the following program.

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

class A {
public:
    A() { cout << "A constructor called" << endl; }
    ~A() { cout << "A destructor called" << endl; }
};

class B {
public:
    B() { cout << "B constructor called" << endl; }
    ~B() { cout << "B destructor called" << endl; }
};

int main() {
    B b;
    static A a;

    return 0;
}
```

What will be the output of the following code?

- a) B constructor called
  A constructor called
  A destructor called
  B destructor called

- b) B constructor called
  A constructor called
  B destructor called
  A destructor called

- c) A constructor called
  B constructor called
  B destructor called
  A destructor called

- d) A constructor called
  B constructor called
  A destructor called
  B destructor called

No, the answer is incorrect.
Score: 0
Accepted Answers:
b) B constructor called
  A constructor called
  B destructor called
  A destructor called
3) Consider the following program.

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

class myClass {
    int data_

public:
    myClass(int d) : data_(d) {} 

    myClass(myClass& m) : data_(m.data_) {
        cout << "Copy constructor called. Data: " << data_ << endl;
    }

    void print() { cout << "Data: " << data_; }

    myClass& operator++() {
        ++data_; 
        return *this;
    }

    myClass operator++(int) {
        myClass t(*this);
        ++data_; 
        return t;
    }

    myClass& operator++(myClass m) {
        data_ += m.data_; 
        return *this;
    }
};

int main() {
    myClass A(10), B(5);

    myClass C = A++ + ++B;

    C.print();

    return 0;
}
```

What will be the output?
a) Copy constructor called. Data: 10
   Copy constructor called. Data: 16  
   Data: 16
b) Copy constructor called. Data: 10
   Copy constructor called. Data: 17  
   Data: 17
c) Copy constructor called. Data: 6
   Copy constructor called. Data: 10  
   Data: 16
d) Copy constructor called. Data: 6
   Copy constructor called. Data: 10
   Copy constructor called. Data: 16  
   Data: 16

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

d) Copy constructor called. Data: 6
   Copy constructor called. Data: 10
   Copy constructor called. Data: 16  
   Data: 16
4) Consider the following program.

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

class Test {
    static int i;
public:
    void setValue(int x) { i = x; }
    int getValue() { return i; }
};

int Test::i = 5;

void fun() {
    Test t1;
    t1.setValue(10);
    cout << t1.getValue() << " ";
    return;
}

int main() {
    Test t;
    fun();
    cout << t.getValue() << endl;
    return 0;
}
```

What will be the output/error?

- a) 5 10
- b) 10 5
- c) 10 10
- d) Error: Undefined reference to Test::i

No, the answer is incorrect.
Score: 0
Accepted Answers:
- c) 10 10
What will be the output of the following program.

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

namespace name1 {
    int x = 5;
}

namespace name2 {
    int x = 10;
}

int main() {
    int x = 20;
    using namespace name2;
    cout << x;
    return 0;
}
```

- a) 5
- b) 10
- c) 20
- d) Error: reference to x is ambiguous

No, the answer is incorrect.
Score: 0
Accepted Answers:
- c) 20

6) 2 points
Consider the following program.

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

namespace {
    int var = 10;
}

int main() {
    int var = 5;
    cout << var;       // LINE-1

    return 0;
}
```

**LINE-1** generates an output as 5. What is/are the correct option/s to change the **LINE-1** so that it will print as 10?

- a) `cout << ::var;`
- b) `cout << namespace::var;`
- c) can’t be accessed the namespace variable `var`.  
- d) `cout << using namespace var;`

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

- a) `cout << ::var;`
7) Consider the program below.

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

class Test {
    static int X;
    public:
        static void Print() {
            cout << " " << X;
        }
};

int Test::X = 10;

int main() {
    Test t;
    t.Print();
    Test::Print();
    return 0;
}
```

What will be the output/error?

- a) 10 10
- b) 10 0
- c) 0 10
- d) Error: Cannot call member function without object

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

8)
Consider the program below.

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

class Singleton {
    int X;
    static Singleton *instance;
    Singleton(int i) : X(i) { }
public:
    int getVal() { return X; }
    static Singleton *createInstance() {
        if (!instance) {
            instance = new Singleton(5);
            cout << "Object Created" "endl;
        }
        else {
            cout << "Already Created" "endl;
        }
        return instance;
    }
};
Singleton *Singleton::instance = 0;

void foo() {
    Singleton *s = Singleton::createInstance();
}

void fun() {
    Singleton *s = Singleton::createInstance();
}

int main() {
    foo();
    fun();

    Singleton *s1 = Singleton::createInstance();

    return 0;
}
```

What will be the output?

a) Object Created
b) Already Created
   Already Created
b) Already Created
   Already Created
   Already Created

c) Object Created
d) Object Created
   Object Created
   Object Created

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

a) Object Created
   Already Created
   Already Created

9) Consider the program below.

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

namespace space {
    int x = 1;
}

namespace space {
    int y = 5;
}

int main() {
    space::x = space::y = 2;
    cout << space::x << space::y;
    return 0;
}

What will be the output/error?

a) 15
b) 22
c) Error: redefinition of namespace is not allowed.
d) 12

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

b) 22