

Some of these questions are taken from last year's exam either directly or with some changes and some are new (i.e., didn't appear in last year's exam).

These sample questions are indicative only, in terms of style and format of the MCQs you will likely solve in this year's exam. Therefore, they are **not** intended to be comprehensive. All material introduced in lectures may be examined unless indicated otherwise.

LIC for COMP6771

Each question is worth **1 mark** with no negative marking. Choose **exactly** one answer to each question.

1. Consider the following C++ program:

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

int main() {
    int a[] = { 1, 2 };
    int *r = a;
    int **p = &r;
    int *&q = *p;
    q = q + 1;
    cout << **p << " " << *q << endl;
}
```

Which of the following about this program is true?

- A. The program compiles and outputs 1 1 when executed
- ☒ B. The program compiles and outputs 1 2 when executed
- C. The program compiles and outputs 2 1 when executed
- ☒ D. The program compiles and outputs 2 2 when executed
- E. The program does not compile

2. What is the output of the following program?

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

class X {
public:
    X() { cout << "ctor "; }
    X(const X&) { cout << "copy-ctor "; }
    ~X() { cout << "dtor "; }
};

int main() {
    list<X**> l;
    X *x1 = new X();
    X** x2 = &x1;
    l.push_back(x2);
}
```

- ☒ A. ctor
- ☒ B. ctor copy-ctor dtor dtor
- C. ctor ctor dtor
- D. ctor dtor
- E. The program does not compile

3. What is the output of the following program?

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

class B {
public:
    virtual void f() { cout << "B::f() "; }
    void g() { cout << "B::g() "; }
};

class D : public B {
public:
    virtual void f() { cout << "D::f() "; }
    void g() { cout << "D::g() "; }
};

int main() {
    B *o1 = new D();
    o1->f();
    o1->g();
    B o2 = *o1;
    o2.f();
}
```

- A. D::f() B::f() D::f()
- ☒ B. D::f() B::g() B::f()
- C. D::f() D::g() D::f()
- ☒ D. D::f() D::g() B::f()
- E. B::f() B::g() B::f()

4. Consider the following program concerning smart pointers:

```

#include <iostream>
#include <string>

using namespace std;

class Y {
public:
    string* operator->() { return new string("abc"); }
};
class X {
public:
    Y* operator->() { return new Y(); }
};
int main() {
    X x;
    cout << x->size() << endl;
}

```

Which of the following statements is true about this program?

- ☒ A. The program does not compile because the operator `*` is not overloaded
- ☐ B. The program does not compile because the member named `size` is not defined in Class `Y`
- ☐ C. The program compiles and the output is 3
- ☐ D. The program compiles but throws a runtime exception since `size` is not defined in Class `Y`
- ☐ E. None of the above

5. What is the output of the following program?

```

#include <iostream>
using namespace std;

class A1 {
public:
    ~A1() { cout << "~A1 "; }
};

class A2 {
public:
    ~A2() {
        cout << "~A2 ";
    }
private:

```

```

    A1 a1;
};

class A3 {
public:
    ~A3() {
        cout << "~A3 ";
    }
private:
    A1 a1;
    A2 a2;
};

```

```

int main() {
    A3 a3;
}

```

- ☒ A. ~A3 ~A2 ~A1 ~A1
- ☐ B. ~A1 ~A1 ~A2 ~A3
- C. ~A3 ~A1 ~A1 ~A2
- D. ~A1 ~A1 ~A3 ~A2
- E. None of the above

6. Consider the following program:

```

#include<iostream>
#include<vector>
using namespace std;

void f() {
    vector<int> v1(10);
    vector<int> *v2 = new vector<int>(10);
    throw 100;
    cout << "never executed!";
}

int main() {
    try { f(); }
    catch (int) { }
}

```

Which of the following about this program is true?

- ☒ A. throw 100 is not allowed since 100 is not a class object

- B. The destructor for `v1` is not called
- C. The destructor for `v2` is not called
- D. The program compiles and the output is 200
- E. All of the above

7. What is the output of the following program?

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

template <typename T> class BigNumber {};

template <typename T> T foo(T) {
    cout << "1 ";
}

template <typename T> BigNumber<T> foo(BigNumber<T>) {
    cout << "2 ";
}

template<> double foo<double>(double) {
    cout << "3 ";
}

int main() {
    BigNumber<double> z;

    foo(2);
    foo(2.0);
    foo(z);
}
```

- A. 3 3 2
- ☒ B. 1 3 2
- C. 1 2 3
- D. 1 1 2
- E. 1 1 1

8. What is the output of the following program?

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

```

class Base {
public:
    virtual void f(float) { cout << "Base::f(float)\n"; }
};
class Derived : public Base {
public:
    virtual void f(int) { cout << "Derived::f(int)\n"; }
};
int main() {
    Derived *d = new Derived();
    Base *b = d;
    b->f(3.14F);
    d->f(3.14F);
}

```

- A. Base::f(float) Derived::f(int)
- B. Base::f(float) Base::f(int)
- C. Derived::f(float) Derived::f(int)
- D. Derived::f(float) Base::f(int)

☒ E. The program does not compile

9. Which of the following about class templates in C++ is true?

- A. A class template cannot have default arguments
- B. Partial template specialisations are not supported
- C. A class template cannot inherit from a non-template class
- D. A class template cannot have a non-template class as its friend

☒ E. None of the above.

10. Consider the following program:

```
1 #include<list>
2
3 using namespace std;
4
5 int main( ) {
6     list<int> *v1 = new list<int>();
7     list<int> *&v2 = v1;
8     list<int> *&&v3 = std::move(v2);
9 }
```

Which of the following statements is true about the two declarations in lines 8 and 9?

- A. There are no compile-time errors.
- B. There is a compile-time error in line 7 only.
- C. There is a compile-time error in line 8 only.
- D. There are compile-time errors in lines 7 and 8.
- ☒ E. None of the above

11. Consider the following program:

```
1 #include<iostream>
2 #include<list>
3
4 int main( ) {
5     int i1 = 1;
6     const int i2 = 2;
7     const int *i3;
8     auto j1 = i1;
9     auto j2 = i2;
10    auto j3 = i3;
11 }
```

The inferred types for j1, j2 and j3 are:

- ☒ A. int, const int, const int *
- B. int, int, int *
- C. int, int, const int *
- D. int, int, const int
- E. int, int, int