

Student Number:

Name:

Write down answers in-between questions. Please answer using short sentences.**The back of each page can be used for practice, but DO NOT write down the answer on the back.****Be sure to write your student number and name on each page.**

1. (12 pts) True / False questions:

- 1) (For most compilers that use "vtable" mechanism for dynamic bounding,) An instance of any C++ class has a hidden member variable named "vptr" or similar. ()
- 2) A vptr of an instance points to the vtable of its declared type. ()
- 3) Static members can only be accessed by class name, not object name. ()
- 4) Function overloading is the redefinition of base class function in its derived class. ()
- 5) If a function t() is declared as a friend function in the class A, t() can directly access private variables in the class A. ()
- 6) If a function t() is declared as a friend function in the class A, an object o in the class A can invoke t() as follows: o.t(); ()

2. (6 pts) Write down the expected output of the following programs. (If a compile error is expected, just write down "error").

```
#include<iostream>
using namespace std;
class Base {
public:
    void f() {
        cout<<"Base\n";
    }
};
class Derived:public Base {
public:
    void f() {
        cout<<"Derived\n";
    }
};
int main() {
    Base *p = new Derived();
    p->f();
    return 0;
}
```

1)

Student Number:

Name:

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

class P {
public:
    void print() { cout <<"Inside P"; }
};

class Q : public P {
public:
    void print() { cout <<"Inside Q"; }
};

class R: public Q { };

int main(void)
{
    R r;
    r.print();
    return 0;
}
```

2)

3. (4 pts) Write down the expected output of the following programs. The position of newline in the answer must also be correct

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

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

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

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

int main() {
    {
        Child child;
        cout << endl;
    }
    cout << endl;
    {
        Test test;
        cout << endl;
    }
    cout << endl;
    return 0;
}
```

1)

Student Number:

Name:

4. (8 pts) Write down the expected output of this program (If a compile error is expected, just write down "error").

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

class Point
{
private:
    double x, y;

public:
    Point(double x_, double y_):x(x_), y(y_) {}

    Point& operator++()
    { x++; y++; cout << "a "; return *this; }

    Point operator++(int)
    { Point temp(*this); x++; y++; cout << "b "; return temp; }

    friend ostream& operator<<(ostream& os, const Point& p);
};

ostream& operator<<(ostream& os, const Point& p)
{
    os << "(" << p.x << ", " << p.y << ")";
    return os;
}

int main(int argc, char* argv[])
{
    Point p1(1, 2);
    cout << ++p1 << endl;

    Point p2(1, 2);
    cout << p2++ << endl;

    return 0;
}
```

5. (6 pts) What is the expected output of the following program?

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

template <typename T>
T add(T a ,T b)
{
    return a + b;
}

int main()
{
    double a = 3.1;
    int b = 5;
    cout << add(a, b) << endl;
    return 0;
}
```

1)

Student Number:

Name:

- 1) 8
- 2) 8.1
- 3) A compile error occurs
- 4) None of the above

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

template <typename T>
void Swap(T& a ,T& b)
{
    cout << "Swap(T& a ,T& b) invoked" << endl;
    T tmp = b;
    b = a;
    a = tmp;
}

template <typename T>
void Swap(T* a ,T* b)
{
    cout << "Swap(T* a ,T* b) invoked" << endl;
    T tmp = *b;
    *b = *a;
    *a = tmp;
}

int main(){
    int a = 3, b = 5;
    Swap(a, b);
    return 0;
}
```

2)

- 1) A compile error occurs.
- 2) A runtime error occurs.
- 3) "Swap(T& a ,T& b) invoked" is printed.
- 4) "Swap(T* a ,T* b) invoked" is printed.

Student Number:

Name:

6. (8 pts) Modify the following code (only one place) to compile successfully and output the following output (The line number is not included in the source code).

```

1. #include <iostream>
2. using namespace std;
3.
4. class A
5. {
6. protected:
7.     int memberA;
8. public:
9.     A(int n):memberA(n) { }
10.    int getMember() { return memberA; }
11. };
12.
13. class B: public A
14. {
15. public:
16.     B() {}
17.     int getMember() { return memberA; }
18. };
19.
20. int main()
21. {
22.     A a(1);
23.     cout << a.getMember() << endl;
24.
25.     B b;
26.     cout << b.getMember() << endl;
27.
28.     return 0;
29. }

```

(output)

```

1
100

```

7. (10 pts) In general, an operator whose result is a new object or the copy of this object returns the new or copied object by value and whose result is an existing, but modified this object returns the reference to the modified this object, although the C++ language rarely puts constraints on operator overloading. Fill in blanks (a)-(f) with the proper return type of the operators based on this policy, to make them behave as similar as possible to the built-in operators.

```

class A {
    __ (a) __ operator =(const A& a);
    __ (b) __ operator +(const A& a) const;
    __ (c) __ operator -() const;
    __ (d) __ operator +=(const A& a);
    __ (e) __ operator ++();
    __ (f) __ operator ++(int);
};

```

Student Number:

Name:

8. (8 pts) Describe 1) where and 2) why a compile error occurs, and 3) modify the code to compile successfully.

```
1. #include <iostream>
2. using namespace std;
3.
4. class A
5. {
6. protected:
7.     int memberA;
8. public:
9.     A(int n):memberA(n) { }
10.    int getMemberA() const
11.    { return memberA; }
12. };
13.
14. class B: public A
15. {
16. private:
17.     const int memberB;
18. public:
19.     B(int n):A(n/2), memberB(n) { }
20.     int getMemberB() { return memberB; }
21. };
22.
23. int main()
24. {
25.     const B b(20);
26.     cout << b.getMemberA() << endl;
27.     cout << b.getMemberB() << endl;
28.
29.     return 0;
30. }
```

1)

2)

3)

Student Number:

Name:

9. (6 pts) What is the expected output of the following program?

```
#include<iostream>
#include<string>
using namespace std;
void f() {
    throw string("error occurred");
    return;
}
int main(){
    try{
        f();
    }
    catch(const char* e)
    {
        cout << "caught an err!\n";
    }
    return 0;
}
```

1)

- 1) "caught an err!" is printed.
- 2) "error occurred" is printed.
- 3) A compile error occurs.
- 4) Program termination due to uncaught exception.

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

void f() {
    throw string("error occurred");
    return;
}
int main(){
    try{
        f();
    }
    catch(...)
    {
        cout << "caught an err!\n";
    }
    return 0;
}
```

2)

- 1) "caught an err!" is printed.
- 2) "error occurred" is printed.
- 3) A compile error occurs.
- 4) Program termination due to uncaught exception.

Student Number:

Name:

10. (8 pts) List all the functions whose addresses are in the vtable of the CSSStudent class.

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

class Person
{
public:
    virtual void talk();
    virtual void getName();
    void sayHi();
};

class Student : public Person
{
public:
    virtual void talk();
    virtual void study();
};

class CSSStudent : public Student
{
public:
    virtual void talk();
    virtual void writeCode();
};

int main()
{
    return 0;
}
```

Student Number:

Name:

11. (12 pts) Fill in the blank (a) appropriately so that you get the following results when you compile and run the following C++ program.

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

class Point
{
public:
    (a)
};

int main()
{
    Point p1(1.1, 2.2);
    cout << "p1 " << p1.x << " " << p1.y << endl;

    Point p2;
    cout << "p2 " << p2.x << " " << p2.y << endl;

    Point p3 = p1;
    cout << "p3 " << p3.x << " " << p3.y << endl;

    return 0;
}
```

```
// execution result

p1 1.1 2.2
p2 3.1 6.2
cctor p3 1.1 2.2
```

Student Number:

Name:

12. (12 pts) Write down the outputs of the following program.

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

class Shape {
public:
    Shape() { cout << "shape created.\n"; }
    virtual ~Shape() { cout << "shape destroyed.\n"; }
    virtual void Draw() const = 0;
    void Print() const { cout << "print shape\n"; }
};

class Circle : public Shape {
public:
    Circle(int r) : Shape(), radius_(r) { cout << "circle created.\n"; }
    virtual ~Circle() { cout << "circle destroyed.\n"; }
    virtual void Draw() const { cout << "draw circle (rad=" << radius_ << ")\n"; }
    void Print() const { cout << "print circle (rad=" << radius_ << ")\n"; }
private:
    int radius_;
};

class Square : public Shape {
public:
    Square(int sz) : Shape(), size_(sz) { cout << "square created.\n"; }
    virtual ~Square() { cout << "square destroyed.\n"; }
    virtual void Draw() const { cout << "draw square (size=" << size_ << ")\n"; }
    void Print() const { cout << "print square (size=" << size_ << ")\n"; }
private:
    int size_;
};

int main() {
    Shape* shapes[2];
    shapes[0] = new Circle(3);
    shapes[1] = new Square(5);
    for (int i = 0; i < 2; ++i) shapes[i]->Print();
    for (int i = 0; i < 2; ++i) shapes[i]->Draw();
    for (int i = 0; i < 2; ++i) delete shapes[i];
    return 0;
}
```