Part I. Answer each questions by choosing A, B, C or D. (40 points)

Note: Please write correct answer in the table.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20

1. _____ is **NOT** member function of a class.
   a) Constructor function
   b) Destructor function
   c) Friend function
   d) Constant function

2. _____ identifies a changeable pointer to constant double data.
   a) const double&
   b) double& const
   c) const double*
   d) double* const

3. The description of _____ is **WRONG**. 啥是静态成员函数
   a) The static member function can be overloaded.
   b) The public static member variable can be called with object.
   c) The public static member variable can be called with class.
   d) The public static member variable can be initialized in constructor function.

4. The description of _____ is **WRONG**.
   a) A friend function can be define anywhere in the class.
   b) A friend function can be inherited.
   c) A friend function can be realized in the class.
   d) A friend function can access to the private members of a class.

5. Derived classes **DON'T** inherit _____of base class.
   a) constructors
   b) virtual functions
   c) static member functions
   d) constant member functions

6. In order to realize polymorphism at running time, _____ must be used.
   a) constructor and destructor
   b) static member functions
   c) constant member functions
   d) virtual functions

7. Which statement is **TRUE** about pure virtual functions?
   a) Pure virtual functions can be inherited.

b) Pure virtual functions contain codes.
c) Pure virtual functions can be realized in derived class.
d) None of above.

8. Which members of the base class can be accessed by the object of a derived class?_____.
a) All members by public inheritance.
b) No members can be accessed.
c) Protected and public members by public inheritance.
d) Only public members by public inheritance.

9. The initialization list of a constructor can not be used to initialize_____.
(a) static data members
(b) references data members
(c) constant data members
(d) member objects

10. Using a virtual base class is in order to _____ .
a) simplify a program.
b) avoid ambiguity.
c) increase the efficiency.
d) reduce the object codes.

11. Here is a piece of code. _____ is correct.
class myclass { const int myVar; };
a) myclass( ) { myVar = 5; }
b) myclass( ) : myVar(5) { }
c) myclass(int a) { myVar = a; }
d) myclass(int myVar) { }

12. _____ will be printed on the screen after running:
int i = 100, &X = i, *p = &X;
i++;
cout << p << "—>" << X;
a) MemoryAddress —> 100
b) MemoryAddress —> 101
c) 100—> 100
d) 101—> 101

13. _____ is correct.
a) int &prt; prt = new int[5]; delete[] prt;
b) int &prt; prt = new int[5]; delete prt;
c) int *prt; prt = new int[5]; delete[] prt;
d) int *prt; prt = new int[5]; delete prt;

14. If **Sample** is a class, _____ is correct copying constructor of this class.
    a) void Sample(const int& a);
    b) void Sample(const int& a) const;
    c) Sample(const int& a);
    d) Sample(const int& a) const;

15. **Sample** is a class, and here is a piece of code: Sample S1, S2(2), S3[3], *S4, &S5(S1). Constructor of **Sample** is called _____ times.
    a) 5 b) 6 c) 7 d) 8

16. When operator ***NEW*** is used, _____ are/is called.
    a) constructor function
    b) member function
    c) destructor function
    d) constructor and destructor functions

17. Where is the assignment operator used? Here myclass is a class.
    a) int x, y = x; b) myclass x, y = x;
    c) myclass x, y(x); d) myclass x, y; x = y;

18. Sample is a class. _____ is used identically as codes:
int x=3; Sample S; S = x.
    a) Sample S = x;
    b) S,operator = ();
    c) S.operator = (S);
    d) S.operator = (Sample(x));

19. What means "**double Add() const**" as a member function of class?
    a) Constant member function is called by constant objects.
    b) Constant member function is called by any objects.
    c) Function, ***Add***, can be realized out of the class.
    d) All above are wrong.

20. Here is a piece of code. _____is right for the source code:

```
class Sample    {
public:    Sample(int a) { X = a; }
           void memberFunction() { cout << X << endl; }
           int X;
};
```

    a) Sample S1;
    b) Sample::memberFunction();
    c) Variable X must be defined in private.
    d) All above are wrong.

Part II. Fill the blanks or write the outputs. (33 points)

1. (9 points)

```
#include <iostream>
using namespace std;
class blah{
public:
   static int a;
   int b;
   blah(int x) {
      b=x;
      a=b+1;
   }
};
int blah::a = 0;
void main(void)
{
   blah b1(5);
   blah b2(12);
   cout << "b1.a is " << b1.a << endl;
   cout << "b1.b is " << b1.b << endl;
   cout << "b2.a is " << b2.a << endl;
   cout << "b2.b is " << b2.b << endl;
}
```

Here gives the outputs:

_____

_____

_____

_____

_____

2. (11 points)

```
#include <iostream>
using namespace std;
class Sample
{
public:
    Sample() {    cout << "Default Constructor is called." << endl;    }
    Sample(const Sample& S){ cout << "Copy onstructor is called." << endl; }
    ~Sample() { cout << "Destructor is called." << endl;    }
};
Sample Fun() { Sample S;    return S;}
void main()
{
     Sample S = Fun();
}
```

Here gives the outputs:

_____

_____

_____

_____

_____

3. (13 points)

```
#include <iostream>
using namespace std;
class Function
{
public:
    Function(int a, int b)   { cout << "Constructor is called. ";   A = a;   B = b; }
    int operator()(int x, int y)   { cout << "operator is called. ";   return   A * x + B * y; }
    operator int()  { cout << "int is called. "; return 2 + (*this)(2,5); }
    int f(int x, int y)   { return   A * (x + y) + B; }
private:
    int A, B;
};
void main()
{
    Function f(2,5);
    int i = f;
    cout << "i = " << i << endl;
    cout << "f(2,5) = " << f(2,5) << endl;
}
```

Here gives the outputs:

_____

_____

_____

_____

_____

_____

_____

Part III. Programming. (27 points)

1. **(12 points)** Define a class, ***Point***, with demands as follows:

   ⑴ The class has member data ***X*** and ***Y*** which describe a point's coordinates.

   ⑵ Define a member function ***Distance*** to get the distance(距离) between two points.

   ⑶ Define a friend function ***Distance*** to get the distance between two points.

   ⑷ A programmer can use ***Point*** in main() as follows:

```
void main()
{
    Point  p,  q(2, 3);    // The coordinate of point p is original point.
    cout << p.Distance(q) << endl;
    cout << Distance(p, q) << endl;
}
```

2. Read the following definition carefully, and complete the questions. **(15 points)**

```
/* The Player class declaration.                    */
#include <iostream>
#include <string>
using namespace std;
class Player
{
public:
    // Print player's name.
    virtual string reportPlayerName() const = 0;
    // Player's first move.
    virtual bool play() = 0;
    // Player's subsequent moves.
    virtual bool play(bool last_play) = 0;
};
```

(1) The ***Player*** class declared above is an example of a(n)_____

a). Interface

b). Concrete class

c). Inherited class

d). Derived class

e). Volatile class

(2) Determine which of the following statements would be "valid" and which would be "not valid" within the context of a program file that has access to the complete definition of the Player class.

Player p1; _____

Player *p2; _____

void topLevel(Player& p1, Player& p2); _____

(3) Write the complete definition of a class named ***MyPlayer,*** derived from ***Player***. The function MyPlayer::reportPlayerName() should return your name, the function MyPlayer::play() should return true and the function MyPlayer::play(bool) should return the argument.