

习题一、

一、选择题

01、下列关于 C++函数的叙述中，正确的是( )。

- A)每个函数至少要具有一个参数
- B)每个函数都必须返回一个值
- C)函数在被调用之前必须先声明
- D)函数不能自己调用自己

02、下列关于类和对象的叙述中，错误的是( )。

- A)一个类只能有一个对象
- B)对象是类的具体实例
- C)类是对某一类对象的抽象
- D)类和对象的关系是一种数据类型与变量的关系

03、在 C++中，用于实现运行时多态性的是( )。

- A)内联函数
- B)重载函数
- C)模板函数
- D)虚函数

04、下列关于运算符重载的叙述中，正确的是( )。

- A)通过运算符重载，可以定义新的运算符
- B)有的运算符只能作为成员函数重载
- C)若重载运算符+，则相应的运算符函数名是+
- D)重载二元运算符时，必须声明两个形参

05、对于语句 `cout<<x<<endl`;错误的是描述是( )。

- A)“cout” 是一个输出流对象
- B)“endl” 的作用是输出回车换行
- C)“x” 是一个变量
- D)“<<” 称作提取运算符

06、有如下类声明，则类 MyDERIVED 中保护的数据成员和成员函数的个数是( )。

```
class MyBASE
{ private:  int k;
  public:  void set(int n){ k=n;}
          int get()const{ return k;}  };
```

```
class MyDERIVED: protected MyBASE
{ protected: int j;
public: void set(int m, int n){ MyBASE::set(m); j=n;}
      int get( ) const { return MyBASE::get( )+j; } };
```

A)4 B)3 C)2 D)1

07、已知在一个类体中包含如下函数原型: `Volume operator-(Volume)const;`, 下列关于这个函数的叙述中, 错误的是( )。

- A)这是运算符-的重载运算符函数
- B)这个函数所重载的运算符是一个一元运算符
- C)这是一个成员函数
- D)这个函数不改变类的任何数据成员的值

08、执行如下的程序段后, 输出结果是( )。

```
cout.fill('*');
cout.width(6);
cout.fill('#');
cout << 123 << endl;
```

A)###123 B)123### C)\*\*\*123 D)123\*\*\*

09、在下列函数原型中, 可以作为类 AA 构造函数的是( )。

A)void AA(int); B)int AA(); C)AA(int)const; D)AA(int);

10、下面四个选项中, ( )是用来声明虚函数的。

A)virtual B)public C)include D)using namespace

11、实现运行时的多态性要使用( )。

- A)重载函数
- B)构造函数
- C)析构函数
- D)虚函数

12、要实现动态联编, 必须通过( )调用虚函数。

- A)对象指针
- B)成员名限定
- C)对象名
- D)派生类名

13、以下( )成员函数表示纯虚函数。

- A)virtual int vf(int);    B)void vf(int)=0;  
C)virtual void vf()=0;    D)virtual void vf(int){ }

14、有如下类声明：

```
class XA
{ private: int x;
  public: XA(int n){ x=n; } };
class XB: public XA
{ private: int y;
  public: XB(int a,int b); };
```

在构造函数 XB 的下列定义中，正确的是( )。

- A)XB::XB(int a,int b): x(a),y(b){ }    B)XB::XB(int a,int b): XA(a),y(b){ }  
C)XB::XB(int a,int b): x(a),XB(b){ }    D)XB::XB(int a,int b): XA(a),XB(b){ }

15、在表达式  $x+y*z$  中，+ 是作为成员函数重载的运算符，\* 是作为非成员函数重载的运算符。则 operator+ 有\_\_个参数，operator\* \_\_有参数。

( )

- A)2、2  
B)2、1  
C)1、2  
D)1、1

16、应在下列程序划线处填入的正确语句是( )。

```
class Base
{ public:
  void fun(){cout<<"Base::fun"<<endl;} };
```

```
class Derived:public Base
{ void fun()
{   //显示调用基类的函数
fun()
  cout<<"Derived::fun"<<endl; }   };
```

- A)fun();    B)Base.fun();    C)Base::fun();    D) Base->fun();

二、填空题

17、面向对象程序设计有四个主要特点，即抽象、封装、\_\_\_\_和\_\_\_\_。

- 18、非成员函数应声明为类的\_\_\_\_函数才能访问这个类的 `private` 成员。
- 19、派生类中的成员不能直接访问基类中的\_\_\_\_成员。
- 20、在用 `class` 定义一个类时，数据成员和成员函数的默认访问权限是\_\_\_\_。
- 21、运算符重载函数可能是类的\_\_\_\_函数，也可以是类的\_\_\_\_函数，还可以是普通函数。
- 22、用流对象的成员函数控制输出格式时，用于设置字段宽度的流成员函数的名称是\_\_\_\_，与之作用相同的控制符名称是\_\_\_\_。
- 23、含有纯虚函数的类称为\_\_\_\_。

### 三、程序的填空、程序的执行

- 24、以下程序是定义一个计数器类 `counter`，对其重载运算符“+”，请填空。

```
class counter
{ private: int n;
  public:
    counter() {n=0;}
    counter(int i){n=i;}
    ____ //运算符重载函数

    { counter t; t.n=n+c.n; return t; }
    void disp() {cout<<"n="<<n<<endl;}
};

void main()
{ counter c1(5),c2(10),c3;
  c3=c1+c2;
  c1.disp(); c2.disp(); c3.disp(); }
```

- 25、下列程序的输出结果为 2，请将程序补充完整。

```
class Base
{ public:
  ____ void fun()
  { cout<<1; }
};

class Derived: public Base
{ public:
  void fun( ) { cout<<2; }
};
```

```
int main( )
{ Base *p= new Derived;
  p->fun( );
  delete p;
  return 0; }
```

26、写出执行下面程序的输出结果。

```
void fun(int &x, int y)
{ int t=x; x=y; y=t; }
int main( )
{ int a[2] = {23, 42};
  fun(a[1],a[0]);
  std::cout<<a[0]<<" "<<a[1]<<std::endl;
  return 0; }
```

27、写出执行下面程序的输出结果。

```
class A { public: A() { cout<<"A"; } };
class B { public: B() { cout<<"B"; } };
class C : public A
{ B b;
  public: C() { cout<<"C"; }
};
int main( ) { C obj; return 0; }
```

28、写出执行下面程序的输出结果。

```
class Base
{ private:
  char c;
public:
  Base(char n):c(n){ }
  virtual ~Base(){cout<<C;}
};
class Der:public Base
{ private:
  char c;
public:
  Der(char n):Base(n+1),c(n){ }
  ~Der(){cout<<C;}
};
int main()
{ Der("X"); return 0; }
```

#### 四、编程题

29、编写一个程序计算“三角形、正方形、圆形”三种图形的面积，要求：

- a)抽象出一个基类 `base`；
- b)在其中说明一个虚函数用来求面积；
- c)利用派生类定义“三角形、正方形、圆形”；
- d)编写主函数并测试。

答案一、

#### 一、选择题

01-05 C A D B D

06-10 B B A D A

11-16 D A C B C C

#### 二、填空题

17、继承、多态性

18、友元

19、私有

20、`private` 或私有

21、成员、友元

22、`width/setw`

23、抽象类

#### 三、程序的填空、程序的执行

24、`counter operator + (counter c)`

25、`virtual`

26、23,23

27、ABC

28、XY

#### 四、编程题

29、

```

#include <iostream>
using namespace std;
class base
{ public:
    virtual float area() const

    { return 0; }
};

class triangle: public base
{ protected:
    float bian,height;
public:

    triangle(float a=0, float h=0) { bian=a; height=h; }

    virtual float area() const

    { return bian*height/2; }
};

class square: public base
{ protected:
    float bian;
public:
    square(float a=0) { bian=a; }

    virtual float area() const

    { return bian*bian; }
};

class circle: public base
{ protected:
    float radius;
public:
    circle(float r=0) { radius=r; }

    virtual float area() const

    { return 3.14*radius*radius; }
};

```

习题二、

一、给出一个有理数类，实现有理数的加法运算。请按后列要求作答。

1. 仔细、完整地阅读程序，对应题号填写注释。注释的内容包括两部分：

1.本行的语句是什么；2.简述其功能或特点。例如，注释可以这样写：“友员函数；重载加法运算符(+)。”

（每小题 1 分，共 10 分）

```
#include<iostream.h>
#include<math.h>
#include<iomanip.h>
class Rational
{ private:
    int numerator,denominator;          //(例)数据成员；表示一个有理数的分子、分母。
    int maxcommonFactor(int,int);      //(1)
    void simplify( );                  //(2)
    void DtoF(double,int&,int&);      //(3)
public:
    Rational(int n,int d=1);           //(4)
    Rational(double x=0);              //(5)
    ~Rational();                       //(6)
    void plus(const Rational&);        //(7)
    Rational operator + (const Rational&); //(例)公有成员函数；重载加法运算符(+).
    void showFraction( );              //(8)
    operator double( );                //(9)
    friend ostream &operator<<      //(10)
        (ostream &output, Rational &t);
};
```

2. 对应题号把函数的定义补充完整。（每小题 3 分，共 9 分）

```
int Rational::maxcommonFactor(int a,int b)
    { if(a%b==0)return b; return (11); }
void Rational::simplify( )
    { int temp=maxcommonFactor(abs(numerator),abs(denominator));
      if(temp==1)return; numerator/=temp;denominator/=temp; return; }
void Rational::DtoF(double x,int & n,int & d)
    { double s=fabs(x); d=1; while((s*d-int(s*d))>1e-15)d*=10;n=int(x*d);}
Rational::Rational(int n,int d)
    { if(d==0) { cout<<"错误！分母不能为 0。有理数将置为 0。"<<endl;
      numerator=0; denominator=1; }
      else { (12); }
      simplify(); }
Rational::Rational(double x)
    { DtoF(x,numerator,denominator);simplify(); }
Rational::~~Rational( ) { }
```



```

void Rational::plus(const Rational& addend)
{
    numerator=addend.numerator*denominator+addend.denominator*numerator;
    denominator*=addend.denominator;
    simplify();
}

Rational Rational::operator + (const Rational& t)
{
    Rational temp(0,1);
    temp.numerator=numerator*t.denominator+denominator*t.numerator;
    temp.denominator=denominator*t.denominator;
    temp.simplify(); return temp;
}

void Rational::showFraction( )
{
    if(numerator<0 && denominator<0 || numerator>=0 && denominator>0)
        cout<<abs(numerator)<<"/"<<abs(denominator);
    else cout<<"-"<<abs(numerator)<<"/"<<abs(denominator);}

Rational::operator double()
{
    return (13);
}

ostream &operator <<(ostream &output, Rational &t)
{
    output<<t.numerator<<"/"<<t.denominator; return output;
}

void main()
{
    Rational a;      cout<<double(a);  cout<<endl;
    Rational b(2,0); b.showFraction(); cout<<endl;
    Rational c(3,4); c.showFraction(); cout<<endl;
    Rational d(1.2); d.showFraction(); cout<<endl;
    a=b+c;          cout<<double(a);  cout<<endl;
}

```

3. 根据题目要求，按对应题号作答。（每小题 3 分，共 21 分）

- 写出上述程序的运行结果。  
(14)
- 在上述程序中，使用成员函数重载加法运算符(+). 现假设要改用友员函数重载加法运算符(+), 下面给出了函数体，请写出相应的函数原型(声明)和函数头。  
函数原型 (15)  
函数头 (16)  
函数体 {Rational temp(0,1);  
temp.numerator=t1.numerator\*t2.denominator+t1.denominator\*t2.numerator;  
temp.denominator=t1.denominator\*t2.denominator;  
temp.simplify(); return temp; }
- 在上述程序中，构造函数出现重载，我们按其定义的先后分别称为构造函数 1 和构造函数 2。注意到 main 函数中依次定义了对象 a,b,c,d。请你分别写出这四个对象的初始化各自调用的是哪一个构造函数。  
对象 a 初始化: (17.a); 对象 b 初始化: (17.b);  
对象 c 初始化: (17.c); 对象 d 初始化: (17.d);
- 在上述程序中，请你补充一个复制构造函数的定义，要求用内联函数实现。  
(18)  
复制构造函数和重载赋值运算符函数虽然都是实现数据成员的复制，但执行时机不同。复制构造函数通常用于(19.a)和传递对象参数，当函数返回数据类型时，也要通过复

制构造函数建立临时对象；重载赋值运算符函数用于(19.b)。

- 假设在上述程序的基础上，于 main 函数末尾处添加一行：

```
for(int i=0;i<=1;i++)c.plus(c+d); cout<<double(c)<<','<< double(d)<<endl;
```

写出这段语句的运行结果。

(20)

## 二、给出一些类的定义，按要求作答。

1. （每小题 3 分，共 18 分）

```
//类 A
#include<iostream.h>
class A
{ private: double x;
  public:
    A(double px=1):x(px){ }
    double getx( ) const {return x;}
    void show( ){cout<<"x="<<x<<endl;}
};
//类 B 私有继承类 A，类 B 的定义及 main 函数如下：
class B: private A
{ private: double x;
  public :
    B(double px=2): x(px){ }    //类 B 构造函数定义
    void show( ){cout<<getx( )<<endl;}
};
void main( ) { A a; a.show( ); B b(3); b.show( ); }
```

- 执行上述程序，运行结果为：

(21)

- 若把类 B 构造函数定义改为 B(double px=2):A(px){ }，则程序运行结果为：

(22)

- 改写上述程序中的类 B 构造函数定义 B(double px=2): x(px){ }。要求不使用“初始化式”，但函数的功能不变，且内联函数的形式也不变。按要求，类 B 构造函数定义可以改写成：

(23)

- 有时需要复制 A 类对象的数据去创建 B 类对象，请用内联函数的形式写出类 B 的重载构造函数：

(24)

- 假设在上述 main 函数的末尾添加一句：cout<<b.getx( )<<endl; 因 getx( )私有,不可访问。请在类 B 的定义中作一声明，保持 getx( )仍为公有，使添加的语句可以运行。该声明为：

(25)

- 在类 B 公有段增加一个内联函数 fetch 的定义，其功能是返回 B 类对象数据 x 的平方值。

(26)

2. （每小题 3 分，共 18 分）

下列程序中，类 B1,B2 虚继承类 A，类 C 多继承 B1,B2。按题目要求作答。注意 A 为虚基类，建立间接派生类对象时，只有一个 a 对象的数据成员版本。

```

#include<iostream.h>
class A                                //长度 x,y
{ public:
    double x,y;
    A(double px=1,double py=1):x(px),y(py){ }
    virtual ~A(){ }
    void show(){cout<<"[A ]=>"<<x<<" "<<y<<endl;}
};
class B1: virtual public A             //矩形面积
{ public :
    B1(double px=2, double py=2):A(px,py){ }
    void show(){cout<<"[B1]=>"<<x*y<<endl;}
};
class B2: virtual public A             //梯形面积
{ public:
    double z;
    B2(double px=3, double py=3, double pz=3):A(px,py),z(pz){ }
    void show(){cout<<"[B2]=>"<<0.5*(x+y)*z<<endl;}
};
class C: public B1, public B2          //体积
{ public:
    double h;
    C(double px=4,double py=4,double pz=4,double ph=4):B2(px,py,pz),h(ph){ }
    void show(){cout<<"[C ]=>"<<x<<" "<<y<<" "<<z<<" "<<0.5*(x+y)*z*h<<endl;}
};
void main()
{   A a;   B1 b1;   B2 b2;   C c;
    A* p=&a;      p->show( );
    p=&b1;         p->show( );
    p=&b2;         p->show( );
    p=&c;          p->show( );
}

```

- 写出上述程序的运行结果。

(27)

- 在类 A 的 show 函数之前加上关键字 virtual，再写出上述程序的运行结果。

(28)

- 在(28)题基础上，于原 main 函数的末尾添加 ((B2)c).show( ); 结果将多显示一行：

(29)

- 把类 A 的 show 函数改写成纯虚函数。

(30)

- 若类 A 的 show 函数改写成纯虚函数，原来的 main 函数不能正常运行，为什么？请简单说明理由。

(31)

- 类 A 的 show 函数改写成纯虚函数后，请你对原来的 main 函数作简单的删改，使其可以

输出派生类的数据。

(32)

### 三、给出一些关于使用模板的程序段，按要求作答。

1. 下列程序用函数模板实现两个数据的交换，请把有关的语句填充完整。

(每小题 3 分，共 9 分)

```
#include<iostream.h>
template <(33)>
void swap((34)) {(35)}
void main()
{ int j=1,k=3;
  cout<<"int 数据类型: \n"<<j<<","<<k<<"="<<"; swap(j,k); cout<<j<<","<<k<<endl;
  double x=1.23,y=9.87;
  cout<<"double 数据类型: \n"<<x<<","<<y<<"="<<"; swap(x,y); cout<<x<<","<<y<<endl;
  char p='A',q='B';
  cout<<"char 数据类型: \n"<<p<<","<<q<<"="<<"; swap(p,q); cout<<p<<","<<q<<endl;
}
```

2. 下列程序使用了向量 vector 和算法 sort 实现数组的排序，请把有关的语句填充完整。

(每小题 3 分，共 9 分)

```
#include<iostream>
#include<vector>
#include<algorithm>
using (36);
const int size=10;
void display(vector<int>V,int n)
{ int i;for(i=0;i<size;i++) cout<<(37)<<" "; cout<<endl;}
bool down(int x,int y)
{ return (38);}
void main( )
{ int a[size]={ 10,3,17,6,15,8,13,34,25,2};
  vector<int>V(a,a+size); // 用数组对模板向量赋初值
  cout<<"输出原始数组: \n";
  display(V,size);
  sort(V.begin( ),V.end( )); // 对向量按升序排序
  cout<<"输出升序排列后的数组: \n";
  display(V,size);
  sort(V.begin( ),V.end( ),down); // 对向量按降序排序
  cout<<"输出降序排列后的数组: \n";
  display(V,size);
}
```

### 四、给出一个输入输出流操作的程序段，请把有关的语句填充完整。

(每小题 3 分，共 6 分)

```
#include <iostream.h>
```

```

#include <fstream.h>
#include <stdlib.h>
void main( )
{ ofstream outf; // 建立输出文件流对象
  outf.open("e:\\newfile.dat", (39));
  if(!outf){ cerr<<"error!"<<endl;abort( );} // 调用重载算符函数测试流
  outf<<"This is a file of example.\n"; // 写入一行内容
  outf (40) ;
}

```

## 答案二

### 一、

1. (每小题 1 分, 共 10 分)

- (1) 私有成员函数; 求两个整数的最大公因子。
- (2) 私有成员函数; 分数约简。
- (3) 私有成员函数; 小数转换为分数。
- (4) 构造函数; 参数为分子、分母形式。
- (5) 重载构造函数; 参数为小数形式。
- (6) 析构函数; 对象的释放清理。
- (7) 公有成员函数; 实现加法运算。
- (8) 公有成员函数; 以分数形式输出有理数。
- (9) 类型转换函数; 把分数形式的对象转换成小数。
- (10) 友员函数; 重载插入运算符(<<).

2. (每小题 3 分, 共 9 分)

- (11) maxcommonFactor(b,a%b);
- (12) numerator=n; denominator=d;
- (13) double(numerator)/double(denominator);

3. (每小题 3 分, 共 21 分)

(14) 0

错误! 分母不能为 0。有理数将置为 0。

0/1

3/4

6/5

0.75

(15) friend Rational operator + (const Rational &t1,const Rational &t2);

(16) Rational operator + (const Rational &t1, const Rational &t2)

(17.a)构造函数 2; (17.b)构造函数 1;

(17.c)构造函数 1; (17.d)构造函数 2;

(18) Rational(const Rational& r) {numerator=r.numerator; denominator=r.denominator;}

(19.a)对象初始化 (19.b)程序运行时修改对象的数据

(20) 6.6, 1.2

### 二、

1. (每小题 3 分, 共 18 分)

(21) x=1

- 1
- (22) `x=1`
- 3
- (23) `B(double px=2) { x=px; }`
- (24) `B(const A& a) {x=a.getx();}`
- (25) 在类 B 的 public 段声明 `A::getx;`
- (26) `double fetch(){return x*x;}`
2. (每小题 3 分, 共 18 分)
- (27) `[A ]=>1,1`  
`[A ]=>2,2`  
`[A ]=>3,3`  
`[A ]=>1,1`
- (28) `[A ]=>1,1`  
`[B1]=>4`  
`[B2]=>9`  
`[C ]=>1,1/4/16`
- (29) `[B2]=>4`
- (30) `virtual void show( )=0;`
- (31) 类 A 的 show 函数改写成纯虚函数, 具有纯虚函数的类 A 成为抽象类, 而抽象类是不能建立对象的。原来的 main 函数中, 有语句 “A a;” 即对类 A 建立对象 a, 因此不能正常运行。
- (32) A 成为抽象类后, 原 main 函数应删除对象 a 的有关语句, 才能正常运行。例如, 可改为:
- ```

B1 b1; B2 b2; C c;
A* p=&b1;    p->show();
p=&b2;       p->show();
p=&c;        p->show();

```

### 三、

1. (每小题 3 分, 共 9 分)

- (33) `typename T`
- (34) `T& a,T& b`
- (35) `T x; x=a; a=b; b=x;`

2. (每小题 3 分, 共 9 分)

- (36) `namespace std`
- (37) `V[i]`
- (38) `x>y`

### 四、

- (每小题 3 分, 共 6 分)

- (39) `ios::out`
- (40) `.close()`

### 习题三、

| Exam Mode | Closed-book exam |   |   | copy           |  |
|-----------|------------------|---|---|----------------|--|
| NUM       | 1                | 2 | 3 | TOTAL<br>SCORE |  |
| SCORE     |                  |   |   | CHECK          |  |

**PART A: Choose the Best Answer For the Following Question (40 Points)**

- Which is incorrect about the OOP? ( B )
  - The central idea of OOP is to build programs using software objects;
  - The OOP focuses mainly on the step-by-step procedure as procedure-oriented programming.
  - The OOP offers many advantages: simplicity, modularity, modifiability, extensibility, and so on.
  - The key concept of object orientation is the attachment of procedure to data.
- Which is incorrect about the object? ( B )
  - An object can be considered as a self-contained computing entity with its own data and programming.
  - In OOP, the software objects have nothing to do with the real objects involved in the application area.
  - An object is known as instance of a given class.
  - Objects help achieve data abstraction and encapsulation.
- Which is incorrect about the class? ( D )
  - A class is a blueprint to objects.
  - we use the keyword class to create a class construct.
  - Once a class is declared, the class name becomes a type name and can be used to declare variables.
  - The class is same as the struct ,and there are no different between class and struct.
- Which is not the standard I/O channel? ( D )
  - std::cin
  - std::cout
  - std::cerr
  - stream
- If there are "int a=5,b=3;", the values of a and b is ( ) after execute "!a&&b++;" ( A )
  - 5,3
  - 0,1
  - 0,3
  - 5,4
- If p and q are pointers that point to a same array, which is incorrect operations with them? (D )
  - if(p>=q);
  - p=null;
  - p=q+1;
  - p=p+q;
- If there are "int m=10;",which is correct statement about reference that listed after? ( A )
  - int &x=m;
  - int &y=10;
  - int &z;
  - float &t=&m;
- If there are "int a=3,\*p=a;",then the value of "\*p" is ( D )
  - the address of variable a;
  - nothing;
  - the address of variable of p;
  - 3;
- If a function has no return value, then its return type must be defined as ( )? ( A )
  - void;
  - any type;
  - int ;
  - no return type;
- which is pass by reference in the functions? ( D )
  - he formal parameter is pointer,and the actual argument is address ;

- B. The formal parameter and the actual argument both are variables;  
 C. The formal parameter and the actual argument both are array name;  
 D. The formal parameter is reference, and the actual argument is variable.
11. In the following keywords, which is to declared a public member in a class? ( B )  
 A. private;      **B. public;**    C. protected;    D. friend;
12. Which is not the character of a constructor? ( D )  
 A. The constructor's name is same as the class;  
 B. The constructor can be overloaded;  
 C. The parameters of constructor may have default values;  
 D. The constructor must be specified the return type;
13. Which is character of a destructor? ( A )  
 A. There is only one destructor in a class.  
 B. The destructor name is not same as the class.  
 C. The destructor must be defined in a class body.  
 D. The destructor may have one or more parameters.
14. Which is incorrect in the following descriptions about the member functions of a class? ( A )  
 A. A member function must be an inline function.  
 B. Member functions may be overloaded.  
 C. A function member must has return type excepts for constructor and destructor.  
 D. The parameters of a member function may have default values.
15. If a variable declare in a unattached function, which scope it is? ( C )  
 A. global scope    B. class scope    C. local scope    D. file scope
16. Which is correct in the following descriptions about functions with optional arguments? ( C )  
 A. It is not allowed to set optional arguments.  
 B. Optional arguments can be set only at the definition of the function.  
 C. Optional argument must be grouped at the end of the formal parameter list.  
 D. If there are any optional arguments in a function, all parameters must be optional arguments.
17. Which is not a function's signature? ( D )  
 A. The number of its formal parameters.  
 B. The types of its formal parameters.  
 C. The order of its formal parameters.  
 D. The return type of the function.
18. If there are "int a[]={0,2,3,4,5}", the value of "a-&a[4]" is (    )? ( B )  
 A. 4      B. -4      C. 5      D. -5
19. Which is incorrect in the following description about the new operator? ( D )  
 A. The new operator is used to allocate storage dynamically.  
 B. A pointer created with new operator can be destroyed by delete operator.  
 C. If using new operator to create an object's pointer, the right construct must be called.  
 D. Initializer always can be supplied for data storage allocated by new.
20. Which is incorrect in the following descriptions about explicit type cast? ( D )  
 A. Static\_cast is used to make conversion between related type.  
 B. Reinterpret\_cast is used to conversion between unrelated type.



- C. Const\_cast is used to drop the read-only status of a type.
- D. A type can be converted to any type with explicit type cast.

**PART B: Fill in the blanks (30 Points)**

1. (15 Points)

|                                                                                            |                                                                                                       |                                                           |
|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| <pre>class A { public:     void f1(); protected:     int i1; private:     int j1; };</pre> | <pre>class B : public A { public:     void f2(); protected:     int i2; private:     int j2; };</pre> | <pre>class C : public B { public:     void f3(); };</pre> |
|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|

(1) Which members are accessible for appendant member f2() ?

f1(), i1, i2, j2

(2) Which members are accessible for object objB of class B?

f1(), f2()

(3) Which members are accessible for appendant member f3() ?

f1(), i1, f2(), i2

2. Which function prototypes have the same signature? (15 Points)

int power(int a, int n );

int power( int a, short n );

double power( double a, int n );

double power( double a, const int n );

double power( double a, int n );    double power( double a, const int n );

**PART C: Show precisely the output of the following C++ program segments. (30 Points)**

1. (15 Points)

```
#include <string.h>
```

```
#include <iostream>
```

```
class Appointment
```

```

{
public:
    Appointment() {}
    Appointment( char *b, int y, int m, int d );
    ~Appointment();
    void display();
private:
    char *what;
    int year, month, day;
};

Appointment::Appointment( char *b, int y, int m, int d )
: year(y), month(m), day(d)
{
    what= new char[50];
    strcpy(what, b);
    std::cout<<"constructor is called."<<std::endl;}

Appointment::~~Appointment()
{
    delete [] what;
    std::cout<<"destructor is called."<<std::endl;}

void Appointment::display()
{
    std::cout<<what<<std::endl;
    std::cout<<year<<" "<<month<<" "<<day<<std::endl;}

int main()
{
    Appointment jack("Good morning!" ,2005, 7, 6);
    jack.display();
    return 0;}

```

Answer:

constructor is called.

Good morning!

2005,7,6

destructor is called.

2. (15 Points)

```

#include <iostream>
using namespace std;
int main()
{
    cout<<"12345678901234567890"<<endl;
    int i = 1234;
    cout<<i<<endl;
    cout.width(12);
    cout<<i<<endl;
    cout.width(12);
    cout.fill('*');
}

```

```

    cout.setf(ios::left,ios::adjustfield);
    cout<<i<<endl;
    cout.width(12);
    cout.setf(ios::right,ios::adjustfield);
    cout.precision(5);
    double j=12.3456789;
    cout<<j<<endl;
    cout<<"width: "<<cout.width()<<endl;
    return 0;
}

```

Answer:

12345678901234567890

1234

1234

1234\*\*\*\*\*

\*\*\*\*\*12.346

width: 0