## C++期末复习

## 面向过程部分

## 基本数据类型

###### 标识符

###### 基本数据类型

###### 构造数据类型

###### 常量 变量的定义

以下选项中合法的用户标识符是( )

A、\_2Test

B、long

C、A.dat

D、3Dmax

## 运算符与表达式

###### 算术 + - \* / % ++ --

###### 条件 > >= < <= == !=

###### 逻辑 ! && ||

###### 任意表达式都有逻辑值

###### 非0 true 0 flase

###### 例如：while（！x）{ } while(1){}

只能作用于整型数据的运算符是【】。

A、-

B、%

C、\*

D、/

设x和y均为bool变量，则表达式x&&y

结果为真的条件是（ ）

A、其中一个为假

B、它们均为假

C、其中一个为真

D、它们均为真

正确表示"x是大于5小于10的数"的

表达式是（ ）

A、x>5&x<10

B、x>5||x<10

C、5<x<10

D、x>5&&x<10

## 三种程序结构

###### 选择 if if …else

###### If() …else if ()…else if() …else

###### switch(int/ char){

###### case 常量1：语句；break；

###### case 常量2：语句；break;

###### default：语句；

###### }

###### while do…while for

###### break continue

已知int a=1,b=2,c=3;以下语句执行后

a,b,c的值是( )

if(a>b) c=a; a=b; b=c;

A、a=2, b=3, c=1

B、a=1, b=2, c=3

C、a=2, b=3, c=3

D、a=2, b=3, c=2

下面程序运行结果是( )

#include<iostream>

using namespace std;

int main()

{

int a,b;

for(a=1,b=1;a<=100;a++)

{

if(b>=20)break;

if(b%3!=1)

b-=5;

else

b+=3;

}

cout<<a<<endl;

return 0;

}

A、7

B、8

C、9

D、10

以下不正确的是( )

A、for（；；）表示无限循环

B、for（）表示无限循环

C、语句for（i=0；；i++）表示无限循环

D、while（1）表示无限循环

## 函数

###### 定义 声明 调用

###### 函数的嵌套调用 递归

###### 函数的重载

###### 带有默认参数的函数

以下正确的函数原型(函数声明）是( )

A、void f1(int ,int);

B、void f1(x,y);

C、void f1(int x,y);

D、f1(int x;int y);

#include<iostream>

using namespace std;

int fun(int x,int y)

{ return(x+y); }

int main()

{

int a=2,b=5,c=8;

cout<<(fun((int)fun(a+c,b),a-c))<<endl;

return 0;

}

程序运行后的输出结果是（ ）

A、9

B、编译出错

C、21

D、9.0

若有声明

“long fun(int \*x,int n,long \*s);

int a[4]={1,2,3,4}; long b,c;”，

则以下函数调用形式中正确的是 。

A. c=fun(a,4,b);

B. c=fun(a[ ],4,&b);

C. c=fun(a[4],4,b);

D. c=fun(a,4,&b);

有以下程序

#include <stdio.h>

int fun()

{ static int x=1;

x+=1;return x;

}

main()

{ int i,s=1;

for(i=1;i<=5;i++)

s+=fun();

cout <<s;

}

程序运行后的输出结果是

A）11 B）21 C）6 D）120

## 数组

###### 一维数组

**一维数组的定义，初始化**

**一维数组的引用**

**一维数组作为函数参数**

以下将一维数组ｍ正确初始化为全零的语句是（ ）。

A、int m[10]=(0,0,0,0);

B、int m[10]={};

C、int m[]={0};

D、int m[10]={10\*2};

( )能正确输出结果：C++

A、char s[3]={'C','+','+','\0'};cout<<s<<endl;

B、char s[3]={'C','+','+'};cout<<s<<endl;

C、char s[]="C++";cout<<s<<endl;

D、char s[3]="C++";cout<<s<<endl;

若有说明:int a[10];

则对a数组元素的正确引用是( )

A、a[10]

B、a[3.5]

C、a(5)

D、a[10-10]

若用数组名作为调用函数的实参，则传递给形参的是( )

A、数组元素的个数

B、数组的第一个元素值

C、数组的首地址

D、数组中全部的元素值

已知有声明“char s[20]=”Hello”;，在程序运行过程中，若要想使数组s中的内容修改为“Good”，则以下语句中能够实现此功能的是 。

A．s="Good"; B．s[20]="Good";

C．strcat(s,"Good"); **D．strcpy(s,"Good");**

## 指针与引用

###### 指针的定义

###### 指针和数组的关系

**指针作为函数参数**

以下程序运行时输出结果是 。

void change(int x, int y, int \*z)

{ int t;

t=x; x=y; y=\*z; \*z=t;

}

void main( )

{ int x=18,y=27,z=63;

change(x,y,&z);

cout <<"x=" <<x <<" y=" << y<< " z= " <<z;

}

输出：x=18 y=27 z=18

知识点：考察对指针作为函数参数时，地址传递的理解

## 面向对象部分

## 1.类

###### 类的定义

###### 类成员的访问权限

###### 类的对象

###### 使用对象访问类中的公有成员 对象名.成员名

###### 使用对象指针访问类中公有成员 指针名->成员名

已知类定义如下：

class Test{

public:

void Set(double val);

private:

double value;

};

则类体外对Set成员函数的定义正确的是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

A、void Set(double val) {value=val;}

B、void Test::Set(double val) {val=value;}

C、Test::void Set(double val) {value=val;}

D、void Test::Set(double val) {value=val;}

知识点：如何在类定义之外定义成员函数

对于下面定义的类MyClass, 在函数f( )中将对象成员n的值修改为50的语句(划线处)应该是( ) 。

class MyClass

{

public:

MyClass(int x)

{

n=x;

}

void SetValue(int n1)

{

n=n1;

}

private:

int n;

};

int f( )

{

MyClass \*ptr=new MyClass(45);

\_\_\_\_\_\_\_\_\_\_;

}

A. MyClass(50) B. SetValue(50)C. ptr - > SetValue(50) D.ptr - > n=50

知识点：使用对象指针访问成员函数的方法

## 2.构造函数与析构函数

###### 生成类的对象时会调用构造函数进行对象的初始化

###### 对象被撤销时，会调用析构函数

###### 构造函数和析构函数的特征

###### 构造函数不带参数（又叫做默认或者缺省构造函数）和带有参数的区别

###### 构造函数可以重载

C++中，系统自动为一个类生成默认构造函数的条件是该类没有自定义任何\_\_\_\_\_。

A、构造函数

B、有参构造函数

C、无参构造函数

D、成员函数

有关类性质说法错误的是\_\_\_\_\_。

A、析构函数和构造函数都不能有返回类型

B、一个类可以有多个构造函数，但只有一个析构函数

C、 一个类中不可以声明具有类类型的数据成员

D、不能给析构函数指定参数

假设A是一个类，下面语句“A a1[2]; A &pa = a1;”执行时，系统会调用\_\_\_\_\_次A类的构造函数。

A、0

B、3

C、2

D、1

知识点：生成类的对象时会调用构造函数进行对象的初始化，这里声明了对象数组a1，有两个元素，这两个对象元素进行初始化调用两次构造函数

A &pa = a1;声明对象的引用（别名），没有新的对象产生，不会调用构造函数

通常，拷贝构造函数的参数是( )。

A. 某个对象名 B. 某个对象的引用名

C. 某个对象的成员名 D. 某个对象的指针名

知识点：拷贝构造函数的特征

## 3.继承与多态

###### 继承的意义和表示方法，在派生类中表示继承关系

###### 继承之后，基类成员在派生类中的访问权限

###### 多态的三个要素，学会构造多态

###### 纯虚函数和抽象类的应用是多态最主要的应用场合

纯虚函数的表示 和 抽象类的特征

派生类的成员函数可以直接访问基类的（ ）成员。

A. 所有 B. 公有和保护 C. 保护和私有 D. 私有

如果一个类至少有一个纯虚函数，那么就称该类为（ 　 ）。

A.抽象类 B. 虚函数 C. 派生类 D. 以上都不对