

2022至2023学年第二学期C++程序设计期末考试**单选题（总分：27.00）**

1. 以下函数定义不正确的是_____

A.

```
void func(int a, int b, int c = 0);
```

B.

```
void func(int a = 0, int b = 0, int c = 0);
```

C.

```
void func(int a, int b = 0, int c);
```

D.

```
void func(int a, int b = 0, int c = 0);
```

正确答案: C

2. 给定如下三个类的定义

```
class X{
    void f();
}

class Y{
    int m, n;
    void f();
}

class Z{
    char s;
    double d;
    void f();
}
```

以上类的实例化对象分别为x,y,z，其内存占用分别为_____

A. 0, 8, 9

B. 1, 8, 12

C. 1, 8, 9

D. 1, 8, 1

正确答案: B

3. 给定如下代码

```
class X{
    private:
        int x1;
    protected:
        int x2;
    public:
        int x3;
}

class Z: public X{
    private:
        int z1;
    protected:
        int z2;
    public:
        int z3;
}
```

那么类Z中访问权限为protected的数据成员有_____

A. x2, x3, z2

B. x1, z2

C. x2, z2

D. z2

正确答案: C

4. 表达式(a + b > 10 || b * 2 <= c) && d - 3 == 9, 其中a=5, b=3, c=10, 运算结果是__

A. true

B. false

C. 11

D. 6

正确答案: B

5.

```
#include <iostream>
void foo(int* ptr, int& ref)
{
    *ptr = 10;
    ref = 20;
}
int main()
{
    int num1 = 1;
    int num2 = 2;
    int* ptr = &num1;
    int& ref = num2;
    foo(ptr, ref);
    std::cout << "num1 = " << num1 << std::endl;
    std::cout << "num2 = " << num2 << std::endl;
    return 0;
}
```

根据以上代码，num1和num2分别等于：_____

- A. 10, 20
- B. 1, 20
- C. 10, 2
- D. 1, 2

正确答案: A

6. 以下定义中，将数组元素全部正确定义的是_____

- A. int a[] = {1, 2, 3, 4};
- B. int a[4] = {1, 2, 3};
- C. int a[4] = {1, 2, 3, 4, 5};
- D. int a[4] = {1, 2, 3, 4, 5, 6};

正确答案: A

7. 关于静态成员的说法，正确的选项是_____

- A. 静态成员变量不能通过类的对象进行访问和修改，而是通过类名和作用域运算符来访问
- B. 静态成员函数不能访问类的非静态成员变量
- C. 每个类对象都有独立的静态成员副本
- D. 静态成员可以被继承，并在派生类中进行重写

正确答案: A

8.

```
int x = 10;

int main(){
    int x = 20;
    {
        int x = 30;
        std::cout << "Inner block: " << x << std::endl;
    }
    std::cout << "Main block: " << x << std::endl;    return 0;
}
```

给定以上代码，以下输出正确的是_____

- A. Inner block: 10, Main block 10
- B. Inner block: 30, Main block 30
- C. Inner block: 30, Main block 20
- D. Inner block: 10, Main block 20

正确答案: C

9. 以下函数声明中，不互为重载函数的是_____

A.

```
void func(int a);
void func(char b);
```

B.

```
void func(int a, int b);
void func(int a);
```

C.

```
void func(int &a);
void func(float a);
```

D.

```
void func(int a);
void func(int b);
```

正确答案: D

填空题（总分：13.00）

1. 给定如下代码

```
class A{
    public:
        A();
        ~A();
}

class B{
    public:
        A a;
        B();
        ~B();
}

class C: public B{
    public:
        C();
        ~C();
}
```

创建C类对象时，构造函数的调用顺序分别是_____

- A. A B C
- B. C B A
- C. B A C
- D. C A B

正确答案: A

2.

```
1. #include <iostream>
2.
3. int main()
4. {
5.     int x = 5
6.     int y = 10;
7.
8.     if (x = y){
9.         std::cout << "x is equal to y." << std::endl;
10.    else
11.        std::cout << "y is equal to x." << std::endl;}
12.
13.    for (int i = 0; i < 5; i++)
14.        std::cout << i << " ";
15.
16.    std::cout << std::endl;
17.
18.
19.    for (int i = 1; i <= 5; i++)
20.        sum += i;
21.        std::cout << "Sum: " << sum << std::endl;
22.
23.    return 0;
24. }
```

请找出以上代码的五处错误，列上标号并指出问题所在

正确答案: 5, 缺少分号 | 8, == | 13, 缺少括号 | 19, 缺少括号 | 20, 没定义sum

3. 写出以下变量的内存占用:

short int: _____

float: _____

long float: _____

char: _____

写出以下变量的取值范围：

unsigned char: _____

正确答案: 2 | 4 | 8 | 1 | 0-255

程序片段编程题（总分：60.00）

1. 拷贝运算（分值：12.00）

根据main函数中的已有内容，补全student类的内容，包括默认构造函数、直接构造函数、拷贝构造函数以及赋值运算符的定义。

填补以下代码，使代码可以顺利运行通过。

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

```
class Student {
public:
    string* name_ptr;
    int age;
```

// 补全默认构造函数，注意指针变量的初始化要求

// 补全直接构造函数

// 补全拷贝构造函数，注意采用深拷贝

// 补全赋值运算符的定义，注意采用深拷贝，且需要自赋值检测

};

int main() {

 string name = "Jerry";

 Student s1(&name, 20); // 调用直接构造函数部分

 Student s2 = s1; // 调用拷贝构造函数部分

 Student s3; // 调用默认构造函数部分

 cout << "s1 name_ptr:" << *s1.name_ptr << ",年龄:" << s1.age << endl;

 cout << "s2 name_ptr:" << *s2.name_ptr << ",年龄:" << s2.age << endl;

 s2 = s1; // 赋值运算符部分

 cout << "s1 name_ptr:" << *s1.name_ptr << endl;

 cout << "s2 name_ptr:" << *s2.name_ptr << endl;

 s2 = s2; // 赋值运算符部分

}

参考答案:


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

class Student {
public:
    string* name_ptr;
    int age;

    // 补全默认构造函数，注意指针变量的初始化要求
    Student() {
        name_ptr = NULL;
        age = 0;
    }

    // 补全直接构造函数
    Student(string* name_ptr, int age) {
        this->name_ptr = name_ptr;
        this->age = age;
    }

    // 补全拷贝构造函数，注意采用深拷贝
    Student(const Student& s) { // 深拷贝构造函数
        name_ptr = new string(*s.name_ptr);
        age = s.age;
    }

    // 补全赋值运算符的定义，注意采用深拷贝，且需要自赋值检测
    Student& operator=(const Student& s) { // 浅拷贝赋值运算符
        if (this == &s) { // 自赋值检测
            return *this;
        }
        name_ptr = s.name_ptr;
        age = s.age;
        return *this;
    }
};

int main() {
    string name = "Jerry";

    Student s1(&name, 20); // 调用直接构造函数部分
    Student s2 = s1; // 调用拷贝构造函数部分
    Student s3; // 调用默认构造函数部分

    cout << "s1 name_ptr:" << *s1.name_ptr << ",年龄:" << s1.age << endl;
    cout << "s2 name_ptr:" << *s2.name_ptr << ",年龄:" << s2.age << endl;

    s2 = s1; // 赋值运算符部分
```

```
    cout << "s1 name_ptr:" << *s1.name_ptr << endl;
    cout << "s2 name_ptr:" << *s2.name_ptr << endl;

    s2 = s2; // 赋值运算符部分
}
```

2. 数字比较 (分值: 9.00)

补全下面代码, 令其输出较大的数字。

例如:

输入1:

1

2

3

输出1:

3

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

```
int sort_number(int x, int y, int z){
```

```
}
```

```
int main()
{
    int num1, num2, num3;

    cin >> num1 >> num2 >> num3;

    cout << sort_number(num1, num2, num3) << endl;
}
```

参考答案:

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

int sort_number(int x, int y, int z){
    int max;
    if (x >= y && x >= z) {
        max = x;
    } else if (y >= x && y >= z) {
        max = y;
    } else {
        max = z;
    }
    return max;
}

int main()
{
    int num1, num2, num3;

    cin >> num1 >> num2 >> num3;

    cout << sort_number(num1, num2, num3) << endl;
}
```

3. 循环控制 (分值: 10.00)

输入一个整数num, 要求输出0-100范围内全部可以被num整除的整数的和。

例如:

输入:

1

输出:

5050

输入

2

输出

2550

注意：小于num的数字不应当计入总和之内。

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

```
int main() {
    int num;
    cin >> num;
    int sum = 0;
```

```
    cout << sum;
}
```

参考答案:

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

int main() {
    int num;
    cin >> num;
    int sum = 0;
    for(int i = num; i < 101; i++){
        if (i % num == 0){
            sum += i;
        }
    }
    cout << sum;
}
```

4. 数组 (分值: 9.00)

给定一个包含五个整数的数组，输出其中最大的数字的坐标位置。(从零计数)

例如：

输入：

1 2 4 5 3

输出

3

输入：

5 4 3 0 6

输出：

4

输入：

7 5 3 2 5

输出：

0

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

```
int main() {
    int nums[5];
    for (int i = 0; i < 5; i++) {
        cin >> nums[i];
    }
```

```
    int maxIndex = 0;
```



```
    cout << maxIndex;
```

```
}
```

参考答案:

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

int main() {
    int nums[5];
    for (int i = 0; i < 5; i++) {
        cin >> nums[i];
    }

    int maxIndex = 0;
    for (int i = 1; i < 5; i++) {
        if (nums[i] > nums[maxIndex]) {
            maxIndex = i;
        }
    }
    cout << maxIndex;
}
```

5. 函数重载 (分值: 10.00)

补全下述代码，定义两个名为area的函数，分别用于圆和矩形面积的计算。

注意，面积应当是float。

示例：

输入：

3

4

输出：

28.26

12

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



```
int main() {
    float a, b;
    cin >> a >> b;
    cout << area(a) << endl;
    cout << area(a, b) << endl;
}
```

参考答案:

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

float area(float a){
    return a * a * 3.14;
}

float area(float a, float b){
    return a * b;
}

int main() {
    float a, b;
    cin >> a >> b;
    cout << area(a) << endl;
    cout << area(a, b) << endl;
}
```

6. 访问器和修改器（分值：10.00）

定义一个椭圆类circle，包含长轴、短轴和面积三个私有数据成员。

为该类提供获取面积的访问器，以及设置长轴和短轴的修改器。

使其能够在修改长轴短轴的同时，对面积进行修改。

此处，椭圆的面积计算公式为

面积 = pi * 长轴 * 短轴。

注意访问器与修改器的命名。

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

```
class Ellipse{
    float a, b, area;
public:
    Ellipse(): a(0), b(0), area(0){}
```

```
};
```

```
int main() {
    float a, b;
    cin >> a >> b;
    Ellipse e1;
    e1.set_ab(a, b);
    cout << e1.get_area() << endl;
}
```

参考答案:


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

class Ellipse{
    float a, b, area;
public:
    Ellipse(): a(0), b(0), area(0){}
    float get_area(){
        return area;
    }
    void set_ab(float x, float y){
        a = x;
        b = y;
        area = 3.14 * x * y;
    }
};

int main() {
    float a, b;
    cin >> a >> b;
    Ellipse e1;
    e1.set_ab(a, b);
    cout << e1.get_area() << endl;
}
```