

答案猜测版本

不保证对, by 高宁

选择题

1. A

通常使用 `typename a;` 的语法创建一个变量, 而其命名有着以下硬性要求:

- 变量名由字母 (广义的Unicode字符, 不包括特殊字符)、数字和下划线构成, 数字不能开头。
- 大小写敏感 (大写的 `a` 和小写的 `A` 是两个不同的变量)。
- 不要跟关键字 (有特殊含义的单词, 后面会讲到) 和系统保留字 (如函数、模块等的名字) 冲突。
- 不能重名。

2. B

字符使用单引号包裹, 其中仅包含一个字母或者转义符号。

3. A

由 `float` 到 `int` 的转换遵守舍入, 因此第一个 $a = 9$, 之后第二个 $a = 9 \times 4 = 36$, 第三个 $a = 9.80665 \times 4 = 39$

4. BD

答案出错了! 很惊讶不是吗?

5. B

首先 `b=1` 的返回值为1, 执行三目运算符前半部分, 返回 `a` 的值, 之后 `a` 自加, 于是对于每一个数据的操作为, `a=a+1`, `b=1`, `c=a` (原来的 `a`), 得到的结果为B

6. D

在小作用域里的 `a` 距离 `int a = 2` 更近, 第一个是2, 之后2被销毁, 第二个 `cout` 时仅存在 `int a = 8` 这一个 `a`, 先自加3之后输出11, 第三个的时候先新建 `a`, 之后赋值5, 括号里面返回4, `a` 赋值4, 输出4, 之后4销毁, 还剩下之前的11, 自加12。

7. A

请自行数数。

8. D

概念题。

9. D

概念题，对于指针和数组的语法糖有一些了解即可得出，D不表示任何含义。

10. D

声明多维数组的时候只有第一个维度可以不声明长度。

11. A

不难得到 `c=6+115`，纯送分题。

12. B

A调用的时候应该输入地址而不是值，C是标准的形参，D调用的时候应该输入值而不是地址

13. C

sum加上i的时候i依次为2、3、4、7、8，手写一下每一遍就好了

14. C

感觉挺明显的，实参剩下三个都可以。

15. B

理性讨论，应该是B，但是假如你写的正确，答案应该是 `a=3, b=0`，因为在最外面的一层的 `switch` 中 `case` 之间没有 `break`，这也是出题错误。

16. C

概念题，结构体没有函数之类的，包括构造函数等等，类有非常多的性质。

17. C

概念题，没啥好说的

18. D

构造函数不可以被继承

19. C

在派生类中重新定义虚函数时，不需加关键字 `virtual` 但是必须要和父类中的同名

20. C

纯虚函数才没有函数体实现部分，虚函数和内联函数均可以有。

程序分析与编程

1. 输出为：

```
统计如下：
xiaomin's number
2022888
xiaomin's age
18
```

```
xiaomin's gender
m
xiaomin's score
0--9: 0
10--19: 0
20--29: 0
30--39: 1
40--49: 1
50--59: 0
60--69: 2
70--79: 1
80--89: 3
90--99: 2
```

2. 依次为:

```
void Swap(int* x, int* y);
Swap(&a, &b);
tmp = *a;
*a = *b;
*b = tmp;
```

3. struct Student
struct Student *p = stu;
p
p
tmp

4. 初始化全局对象
进入 main 函数
初始化自动局部对象
调用子函数
进入 function () 函数
初始化自动局部对象
初始化静态局部对象
进入 function () 函数
初始化自动局部对象
main 函数执行结束

```
5. class Line : public Dot
    Input Dot1:
    3 8
    Input Dot2:
    2 6
    Dot1: x=3      y=8
    Dot2: x=2      y=6
    Length=2.23607Center: x=2.5    y=4.5
```

```
6. this->i = a.i + b.i;
    MyClass operator+(const MyClass &a)
    i=10
    i=20
    i=30
```

```
7. A(float a, float b)
    friend float Sum(const A &obj) { return obj.x + obj.y;
    };
    9
    30
```

```
8. template <typename T>
    T findmax(T arr[], int len)
    T val
    findmax<double>(arr2, 3);
```

```
9. template <typename T>
    T
    template <typename T>
    DoubleDArray<T>::DoubleDArray()
```

```
10. #include <iostream>

    class myNeure {
    private:
        double w1, w2, x1, x2, b, f, neureResult;

    public:
        myNeure(double w1_val, double w2_val, double x1_val,
        double x2_val, double b_val)
            : w1(w1_val), w2(w2_val), x1(x1_val), x2(x2_val),
        b(b_val) {
            calculate();
```

```
}

~myNeure() {
    std::cout << "myNeure object is destroyed." <<
std::endl;
}

void calculate() {
    f = w1 * x1 + w2 * x2 + b;
    neureResult = (f > 0) ? f : 0;
}

void print() {
    std::cout << "Neure Result: " << neureResult <<
std::endl;
}
};

int main() {
    myNeure neure1(2.3, 1.8, 0.9, 0.5, 0.003);
    neure1.print();
    return 0;
}
```