**面向对象程序设计作业-第1章 绪论**

|  |  |  |
| --- | --- | --- |
| 姓名： | | |
| 本章教学内容 | | |
| 教学内容 | 课内学时 | MOOC学习内容 |
| 第1章 绪论  1.1课程简介  1.2函数模板【重难点】 | 1  1 | 相应教学内容 |
| 作业及要求 | | |
| 1.结合课本第7章课后练习及MOOC相关内容复习本周教学内容；  2.需提交作业：  （1）基本概念题  （2）简答题  （3）选择题  （4）程序设计题  3.作业提交的相关说明：  （1）按作业模板撰写作业文档；  （2）作业提交方式：请班长/学委收齐后发至邮箱zuozejun@mapgis.com；  （3）作业提交时间：周二（2月19日）晚10:00前；  （4）每次作业根据提交的数量和质量评定成绩。 | | |

## 一、基本概念题

1.C++最重要的特性之一就是代码重用，为了实现代码重用，代码必须具有  通用性 。

通用代码需要不受数据 类型 的影响，并且可以自动适应数据类型的变化。这种程序设计类型称为 参数化 程序设计。模板是C++支持参数化程序设计的工具，通过它可以实现参数化 性。

2.函数模板的定义形式是template<模板参数表> 返回类型 函数名（形式参数表）{…}。

其中，<模板参数表>中参数可以有 多 个，用逗号分开。模板参数主要是 模板类型 参数。

它代表一种类型，由关键字 typename 或 class 后加一个标识符构成，标识符代表一个潜在的内置或用户定义的类型参数。类型参数又可以是任意合法标识符。C++规定参数名必须在函数定义中至少出现一次。

3.编译器通过如下匹配规则确定调用哪一个函数：首先，寻找最符合 函数名 和 参数类型 的一般函数，若找到则调用该函数；否则寻找一个 函数模板 ，将其实例化成一个 模板函数 ，看是否匹配，如果匹配，就调用该 模板函数 ；

再则，通过 类型转换 规则进行参数的匹配。若还没有找到匹配的函数则调用错误。若有多于一个函数匹配，则调用产生 二义性 ，也将产生错误。

4.类模板使用户可以为类声明一种模式，使得类中的某些数据成员、某些成员函数的参数、某些成员函数的返回值能取 任意类型 （包括 系统预定类型 和 用户自定义的类型 的类型）。类是对一组对象的公共性质的抽象，而类模板则是对不同类的 数据类型 的抽象，因此类模板是属于更高层次的抽象。由于类模板需要一种或多种 类型 参数，所以类模板也常常称为 参数化类 。

## 二、简答题

1.简述函数模板生成函数的过程。

编译器检测到调用模板的语句时，进行模板实例化，将函数类型换为需要的类型，生成函数。

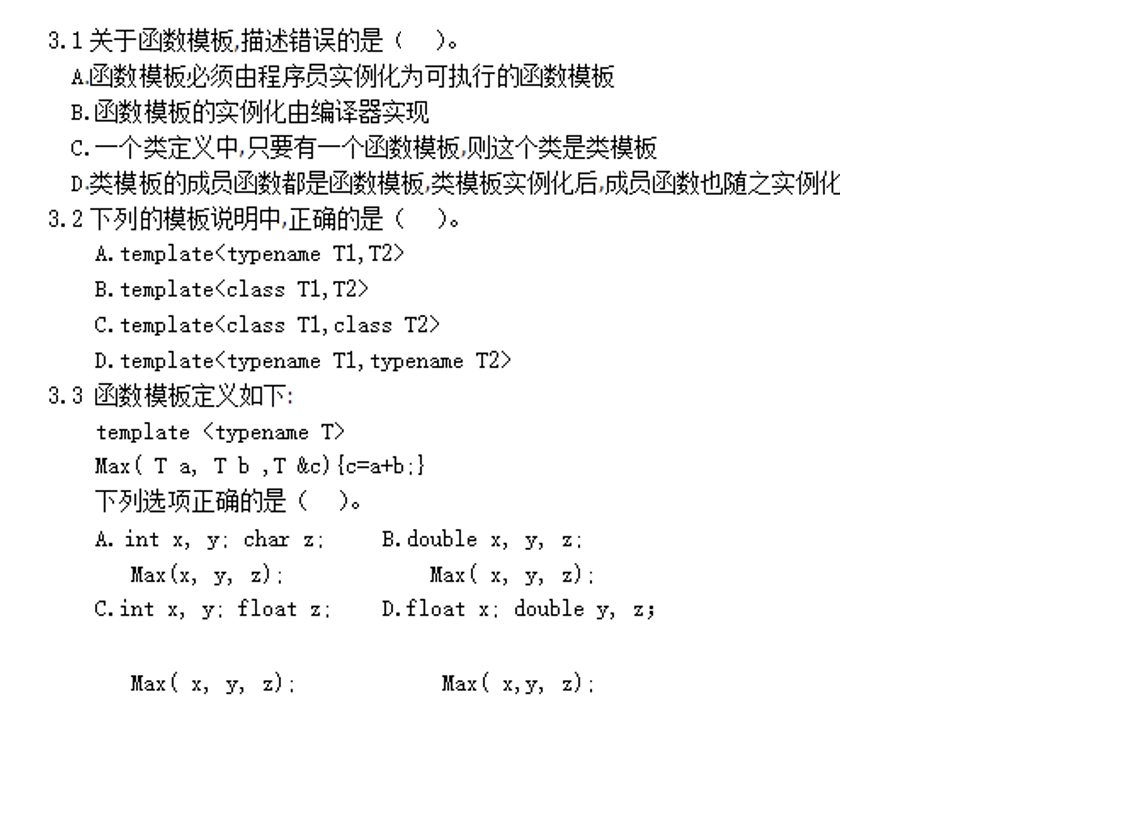
2.简述类模板生成对象的过程。

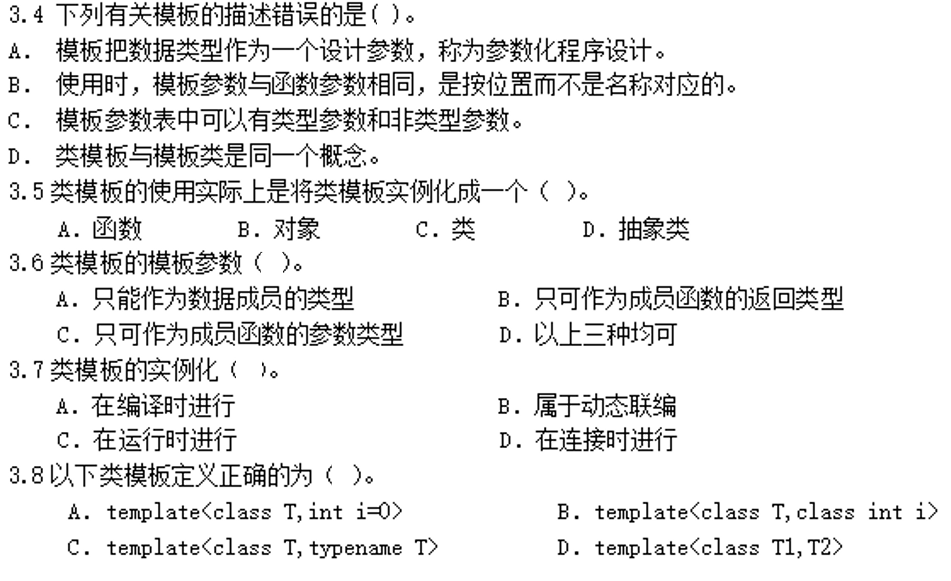
编译器检测到调用模板的语句时，进行模板实例化，将类的成员参数类型换为需要的类型，生成模板类。

3.简述函数模板与模板函数、类模板与模板类的区别。

函数模板只改变函数的成员类型，类模板可以修改成员函数类型

## 三、选择题





答案：ADBDC DAA

## 四、程序设计题

1．设计一个函数模板，其中包括数据成员T a[n]以及对其进行排序的成员函数sort( )，模板参数T可实例化成字符串。

#include<iostream>

#include<string>

using namespace std;

template < typename T, int n>

void sort(T (&arr)[n]) {

T temp;

for(int i=0;i<n;i++)

for(int j=i+1;j<n;j++)

if (arr[i] < arr[j]) {

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

int main() {

int a[4] = { 5,4,1,9 };

sort<int,4>(a);

for (int i = 0; i < 4; i++)

cout << a[i] << endl;

string b[4]={ "asd","asfa","qwfdacac","asfdsfbdfdfasd"};

sort<string, 4>(b);

for (int i = 0; i < 4; i++)

cout << b[i] << endl;

return 0;

}

2．设计一个类模板，其中包括数据成员T a[n]以及在其中进行查找数据元素的函数int search(T)模板参数 T可实例化成字符串。

#include<iostream>

#include<string>

using namespace std;

template < typename T>

class A {

public:

T n[10];

int search(T) {

T t;

cin >> t;

for (int i = 0; i < 10; i++)

if (n[i] == t)

cout << "目标在第" << i + 1 << "个" << endl;

return 0;

}

};

int main() {

int a[10] = { 19,4,86,54,98,74,6,51,56,489 };

A<int>c1;

cout << "请输入要查找的元素" << endl;

c1.search(a[10]);

string b[10] = { "zxc","vbn","mas","dfg","abc","hgw","qre","wtr","ty","uyi" };

A<string>c2;

cout << "请输入要查找的元素" << endl;

c2.search(b[10]);

return 0;

}

## 五、本章学习小结

掌握函数模板和类模板的定义和基本使用方法，在定义类模板时容易犯错。

## 六、本章学习待答疑问题