上 海 交 通 大 学 试 卷（A卷）

（ 2019 至 2020 学年 第 1 学期 ）

班级号 _____ 学号_____ 姓名 _____

课程名称 CS154 程序设计思想与方法（C++） 成绩 _____

一．选择填空题（每题 1 分， 共 20 分）

1. 设有语句 char ch; cin.get(ch); 下列能够输出 ch 的 ASCII 码的语句是__A__。

   A. cout<<(int)ch;　　B. cout<<"ch";　　C. cout<<ch;　　D. cout<<ch+'0';

2. 有如下的枚举定义：

   enum color { Monday=1, Tuesday, Wednesday, Thursday, Friday, Saturday=11, Sunday};

   则 Wednesday 和 Sunday 的值分别为___C_____。

   A. 3, 7　　B. 2, 6　　C. 3, 12　　D. 2, 7

3. 执行下列语句后，s 的值是多少___D_____。

```
int s=0;
for(int i=0;i<=10;i++)
   if(i%2==0 || i%3==0) continue;
   else s+=i;
```

   A.12　　B.11　　C.14　　D.13

4. 有整型变量 y=5; 运行代码的结果是显示: NOT OK. 以下正确的代码块是：__C____。

A.	B.	C.	D.
y=(y>1)?0:10;	if (10>y>2)	switch(y){	while(y>=5)
if (y)	cout<<"NOT";	case 5:	{
cout<<"NOT";	if(2<y<10)	cout<<"NOT";	cout<<"NOT";
cout<<" OK";	cout<<" OK";	default :	}
		cout<<" OK";	cout<<" OK";
		}	

5. 下列循环语句有错误的是___ A _____。

   A. int i = 0, sz = 10; for (i != sz; ++i) {…}　　B. for (int i=0, sz=10; i < sz; ++i) {…}

   C. while (1) {…}　　D. int i = 0, sz = 1; while (i != sz) {…}

6. 以下对跳转语句 continue 和 break 的叙述错误的是___D_____。

   A. continue 语句只能出现在循环内部　　B. break 语句可以用于 switch 语句中

   C. continue 语句终止最近的循环中的当前迭代，并立即开始下一次迭代

   D. 若 while 循环中包含 switch 语句，switch 语句中的 break 可以跳出 while 循环

7. 执行 C++语句 cin.getline(str, 80, '$'); 从输入流读取字符串后，字符数组 str 中不可能包含的字符是__C____。

   A. '\n'　　B. ' '　　C. '$'　　D. '\0'

我承诺，我将严格遵守考试纪律。

承诺人：_____

题号	一	二	三	四
得分				
批阅人(流水阅卷教师签名处)				

8. 已知 char str[10] = "2019\t"；则 strlen(str)+sizeof(str)的值是__B__ 。

A. 小于等于 14　　B. 15　　C. 16　　D. 17

9. 以下有关函数的说法，正确的是__ B ____ 。

A. 不能定义两个函数名一样的函数　　B. 两个函数的原型声明不能相同

C. 函数返回时所有局部变量都会自动被撤销

D. 参数传递是值传递，数组作为参数传递时，数组按元素逐一传递

10. 下面有关变量生命周期的说法，正确的是___C___ 。

A. 自动变量是由系统自动在堆区域为其分配空间，当定义它的函数返回时自动消亡

B. 静态局部变量存在于系统的栈区域，会随着定义的函数返回而消亡

C. 静态局部变量和静态全局变量一样，都是在整个程序退出时才消亡

D. 以上说法都是错误的

11. 已知函数 fun 的原型是：void fun(double *a, char &b,); 变量 v1、v2 的定义是：double v1[100]; char v2;，正确的调用语句是_ B _____ 。

A. fun (v1, &v2) ; B. fun (v1, v2); C. fun (&v1, &v2) ; D. fun (&v1, v2);

12. 若有如下定义，则对数组元素的成员引用不正确的是_C_____ 。

struct Student{int id; char name[20]; char gender; int age;};

Student stu [2],*p; p=stu;

A. p[1].gender　B. p->age　　C. p[0]->id　　D. stu[1].name

13. 当说明一个结构体变量时，系统分配给它的内存是_____A_____。

A. 结构体各成员所需内存量的总和　　B. 结构体中第一个成员所需内存量

C. 结构体成员中占内存量最大者所需内存量　　D. 结构体中最后一个成员所需内存量

14. 能够释放对象所占资源的是___A_____。

A. 析构函数　　B. 拷贝构造函数　　C. 构造函数　　D. 静态成员函数

15. 已知 A 类，则当程序执行到语句：A a[4],*pa[3];时，调用了____B_____次构造函数。

A. 3　　B. 4　　C. 7　　D. 8

16．如果运算符= =被重载为类成员函数，则表达式 obj1 = = obj2 被 C++编译器解释为___C_____。

A．= =(obj1,obj2)　　B. operator= =(obj1,obj2)

C．obj1.operator= =(obj2)　　D. obj2.operator= =(obj1)

17. 如将类 X 的*运算符重载为友元函数，实现类 X 的两个对象相乘，并返回相乘后的结果，则该函数的声明语句为___A_____。

A. X operator*(const X & a , const X & b);　　B. X operator*(const X & a) const;

C. X & operator*(const X & a , const X & b);　　D. X & operator*(const X & a) const;

18．类模板的使用实际上是将类模板实例化成一个具体的___C_____。

A. 对象　　B. 函数　　C. 类　　D. 模板

19. 下列说法中错误的是 ____B____。

A. 公有继承时基类中的 public 成员在派生类中仍是 public 的

B. 公有继承时基类中的 private 成员在派生类中仍是 private 的

C. 私有继承时基类中的 public 成员在派生类中是 private 的

D. 保护继承时基类中的 public 成员在派生类中是 protected 的

20．执行以下语句： char ch[10], ch1[10]; cin.getline(ch, 8, '.'); cin.getline(ch1, 8, '.');

当用户输入：abc.def.（回车）后 cout<<ch1 的结果为：___A_____。

A.　“def”　　B.　“.def”　　C.　“def.”　　D.　“abc.def.”

二．给出下列程序段的运行结果（每题 3 分， 共 15 分）

1.
```
#include <iostream>
using namespace std;
int main(){
    int n=21;
    cout<<n;
    while(n!=1){
        if(n%2==0){
            n/=2;
            cout<<"->"<<n;
        }
        else{
            n=n*3+1;
            cout<<"->"<<n;
        }
    }
    return 0;
}
```
参考答案：21->64->32->16->8->4->2->1

2.

```
#include <iostream>
using namespace std;
int main(){
    for (int i=0; i < 3; ++i){
        int j = 5;
        while(i < j){
            j = j - 3 - i;
            cout << '*';
        }
        switch (j) {
        case -1:
            cout << 'A';
            break;
        case 0:
            cout << 'B';
            break;
        case 1:
            cout << 'C';
        default:
            cout << 'E';
        }
    }
    return 0;
}
```

参考答案：**A*CE*B

3.

```
#include <iostream>
using namespace std;
int *addone(int &arg){
    arg += 1;
    int *ptr = new int(arg);
    return ptr;
}
int main() {
    int a[5] = {1, 2, 3, 4}, *p, i;
    p = a;
    for(i=0; i<5; ++i) {
        p = addone(a[i]);
        a[i] *= *p;
        delete p;
        cout << a[i] << endl;
    }
    return 0;
}
```

参考答案：4 9 16 25 1

gei

4.

```
#include <iostream>
using namespace std;
class CC
{
private:
    int A,B;
public:
    CC(int i,int j)
    {A=i;B=j;cout<<"Call constructor\n";}
    CC(const CC &obj)
    {A=obj.A+1;B=obj.B+2;cout<<"Call copy constructor\n";}
    ~CC()
    {cout<<"Call destructor\n";}
    void print()
    {cout<<"A="<<A<<", B="<<B<<endl; }
};
int main()
{   CC a1(2,3);
    CC a2(a1);
    a2.print();
    CC *p = new CC(5,6);
    p->print();
    delete p;
    return 0;
}
```

参考答案：

```
Call constructor
Call copy constructor
A=3, B=5
Call constructor
A=5, B=6
Call destructor
Call destructor
Call destructor
```

5.

```
#include <iostream>
using namespace std;
class A
{
private:
    int x;
    static int count;
    int sum_count;
public:
    A(int a = 1, int c = 0) :x(a), sum_count(c) { count++; }
    A operator+(A &r)
    {   r.sum_count++;
        sum_count++;
        return A(x + r.x);
    }
```

```
    void disp() { cout << "x: " << x << " count : " << count
    << " sum_count: " << sum_count << endl; }
};

int A::count = 0;
A a0 = 9;

int main()
{
   A a1(2);
   A a2 = a0 + a1;
   a0 = a0 + a2;
   a0.disp(); a2.disp();
   return 0;
}
```

参考答案：

x: 20 count: 4 sum_count: 0

x: 11 count: 4 sum_count: 1

三．程序填空（每空 2 分， 共 30 分）

1. 下面是一个 DoubleArray 类定义和实现，主要完成赋值运算符重载。

```
#include <iostream.h>
class DoubleArray{
private:
    int low;
    int high;
    double *storage;
public:
    // 构造函数根据low和high为数组分配空间
    DoubleArray(int lh = 0, int rh = 0)_____________________
    {   storage = new double [high - low + 1]; }

    DoubleArray &operator=(const DoubleArray &right); // 赋值运算符重载函数
    ~DoubleArray() { if (storage) delete [] storage; } //析构函数
};

DoubleArray &DoubleArray::operator=(const DoubleArray & a)
{
    if (_______________) return _________________;
    delete [] storage;

   low = a.low;   high = a.high;
   storage = ____________________________________________;
  for (int i=0; i <= high - low; ++i) storage[i] = a.storage[i];

   return *this;
}
```

参考答案：

1. :low(lh), high(rh) 2. this==&a 3. *this 4. new double[high-low+1]

2. 下列程序将两个字符串拼接起来。

```
#include <iostream>
using namespace std;
void myStrCat(char *str1,char *str2);
int main(void)
{
    char dst[100] = "Hello, ";
    char src[100] = "Welcome to C++!";
    _____                    //调用myStrCat函数
    cout<<dst<<endl;
    return 0;
}
void myStrCat(char *str1,char *str2)
{
    int i=0, len;
    len = strlen(str1);
    while(_____)
    {
        _____                    //通过赋值，实现str2字符拼接至str1后面
        len++;
        i++;
    }
    str1[len] = '\0';
}
```

参考答案：1. myStrCat(dst, src)　2. str2[i] != '\0'　3.str1[len]=str2[i]

3. 数组中的峰值元素是指其值大于左右相邻元素值的元素。一个数组中可能有多个峰值，如数组元素［1，2，1，3，4，1］中 2 和 4 都是峰值。编程实现数组中所有峰值元素的值。

```
#include <iostream>
using namespace std;
int main()
{   int *arr;
    int cur, num;

    cout << "输入元素个数：";   cin >> num;
    _____
```

```
    for ( int i = 0; i < num; ++i)
        cin >> arr[i];
    cur = 1;

    while (_____) {
        if ((arr[cur-1] < arr[cur])&&(arr[cur+1] < arr[cur]))
            cout << arr[cur] << endl;
        cur++;
    }

    _____
    return 0;
}
```
参考答案：1. arr = new int[num];　2. cur < num　3. delete [] arr;

4. 将 1~100 内所有的偶数写入文本文件 file.txt，然后从 file.txt 中读取这些数据，并显示在屏幕上。

```
#include<iostream>
#include_____
using namespace std;
int main() {
    ofstream out("file.txt");
    ifstream in;
    int i;

    if (_____) { cerr << "creat file error\n"; return 1; }
    for (i = 1; i <= 100; ++i) {
        if (_____)
            out << i << ' ';
    }
    out.close();

    in._____ ("file.txt");
    while (_____) cout << i << ' ';
    in.close();

    return 0;
}
```
参考答案：1. <fstream>　2. !out　3. !(i%2)　4. open　5. in >> i

四、编程题（共 35 分）

1. 编写程序将十进制正整数 d 转化成 k 进制数（2≤k≤16），并输出。例如十进制数 60 转化成十六进制表示为 3C。（5 分）

程序运行示例：(说明：下划线部分是用户输入内容，其他为程序输出内容)

请输入 d 和 k：60 16

3C

参考答案：

```
int main()
{
    char c[40];
    int d, k, i, j;

    cout << “请输入 d 和 k：”;
    cin >> d >> k;
    for (i = 0; d != 0; i++) {
        j = d % k;
       c[i] = (j < 10) ? (j + '0') : (j - 10 + 'A');
        d /= k;
    }
    for (i--; i>=0; i--)    cout << c[i];
}
```

2. 定义一个日期类 TDate，使得下面测试程序（15 分）

```
int main()
{
    TDate dt1(2,28,2020),dt2(12,31,2020);
    dt1.nextDay();
    dt1.print();
    cout << endl;
    dt2.nextDay();
    dt2.print();
    cout << endl;

    return 0;
}
```

运行结果为：

```
2/29/2020
1/1/2021
```

其中：函数 nextDay()实现求第二天日期的功能，函数 print()实现输出日期的功能。

参考答案：

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

class TDate            // 3 分
{
private:
    int month;
    int day;
    int year;
    bool isLoopYear();
public:
    TDate(int,int,int);
    void print() const;
    void nextDay();
};

TDate::TDate(int m,int d,int y)        // 2 分
{
    month = m;
    day = d;
    year = y;
}
```

[键入文字]

```
bool TDate::isLoopYear()      //2 分
{
      return(((year % 4 == 0)&&(year % 100 != 0))||(year % 400 == 0));
}

void TDate::print() const          //2 分
{
      cout << month << '/' << day << '/' << year;
}

void TDate::nextDay()          //6 分
{
      int monthDays[13] = {0,31,28,31,30,31,30,31,31,30,31,30,31};
      if((month == 2)&&isLoopYear())
            monthDays[2] = 29;
      if(day < monthDays[month])
            ++day;
      else
      {
            day = 1;
            ++month;
            if(month > 12)
            {
                  month = 1;
                  ++year;
            }
      }
}
```

说明：如果没有单独写 isLoopYear()函数，而是直接在 nextDay()函数中实现，nextDay()函数满分为 8 分。

3. 实现函数 int* searchRange(int array[], int sz, int target)，该函数完成如下功能：对于一个已经按照非递增顺序排列的长度为 sz 的整型数组 array，寻找包含整数 target 的区间。如果有，返回包含左右边界下标的数组；若没有，返回整型数组[-1, -1]。请使用二分查找分别寻找左右边界。（15 分）

运行下面测试程序

```
int main()
{
   int array[] = {26, 24, 15, 15, 15, 7, 4, 2};
   int target = 15;

   int *p = searchRange(array, 8, target);
   cout << "[" << p[0] <<"," << p[1] << "]" << endl;

   target = 8;
   p = searchRange(array, 8, target);
   cout << "[" << p[0] <<"," << p[1] << "]" << endl;
}
```

结果为：

[2,4]

[-1,-1]

参考答案：

```
int* searchRange(int array[], int sz, int target)
{
      int* result = new int[2];
      int prev_l = 0, l = sz - 1;          // 1 分

      while (prev_l < l)                        //4 分
       {
       int mid = (prev_l + l) / 2;
       if (target < array[mid])
             prev_l = mid + 1;
       else
             l = mid;
      }
      if (array[l] == target)            // 3 分
       result[0] = l;
      else {
       result[0] = -1, result[1] = -1;
       return result;
      }

      int prev_r = sz - 1, r = 0;          //1 分
      while (prev_r > r) {                  //4 分
       int mid = (prev_r + r + 1) / 2;
       if (target <= array[mid])
             r = mid;
       else
             prev_r = mid - 1;
      }
      result[1] = r;                       //1 分

      return result;                       //1 分
}
```