

# Appendix A

## 练习答案

## 第1章 入门的第一步

① ×      ② ○      ③ ×      ④ ×      ⑤ ×

## 第2章 C++ 的基础知识

1. 该代码在语法上没有错误。编译之后可以执行代码。但是该代码非常难以阅读。  
通过添加改行或缩进等可以变成容易阅读的代码，代码如下所示。

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

int main()
{
    cout << " 你好 \n";
    cout << " 再见 \n";

    return 0;
}
```

2.

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

int main()
{
    // 分开显示 123 和 45
    cout << 1 << 2 << 3 << '\n' << 4 << 5 << '\n';

    return 0;
}
```

3.

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

int main()
{
```

```
cout << 123 << '\n';
cout << "收了" << '\¥' << "100\n";
cout << "明天见\n";

return 0;
}
```

4.

## ●八进制

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

int main()
{
    cout << 06 << '\n';
    cout << 024 << '\n';
    cout << 015 << '\n';

    return 0;
}
```

## ●十六进制

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

int main()
{
    cout << 0x6 << '\n';
    cout << 0x14 << '\n';
    cout << 0xD << '\n';

    return 0;
}
```

## 第3章 变量

1.

```
# include <iostream>
```

```
using namespace std;

int main()
{
    double pi;
    cout << " 圆周率的值是多少? \n";
    cin >> pi;
    cout << " 圆周率的值是 " << pi << "。 \n";

    return 0;
}
```

2.

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

int main()
{
    char ch;

    cout << " 字母表的第一个字母是什么? \n";
    cin >> ch;
    cout << " 字母表的第一个字母是 " << ch << "。 \n";

    return 0;
}
```

3.

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

int main()
{
    double height, weight;

    cout << " 请输入身高和体重。 \n";
    cin >> height >> weight;
    cout << " 身高是 " << height << " 厘米。 \n";
}
```

4

```
cout << " 体重是 " << weight << " 公斤。 \n";

return 0;
}
```

Lesson

**A**

## 第 4 章 表达式与运算符

1.

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

int main()
{
    int ans1 = 0-4;
    double ans2 = 3.14*2;
    double ans3 = (double)5/(double)3;
    double ans4 = 30%7;
    double ans5 = (7+32)/(double)5;

    cout<< "0-4 是 " << ans1 << "。 \n";
    cout<< "3.14×2 是 " << ans2 << "。 \n";
    cout<< "5÷3 是 " << ans3 << "。 \n";
    cout<< "30÷7 的余数是 " << ans4 << "。 \n";
    cout<< "(7+32)÷5 是 " << ans5 << "。 \n";

    return 0;
}
```

2.

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

int main()
{
    double height,width;

    cout << " 请输入三角形的高。 \n";
```

```
    cin >> height;
    cout << " 请输入三角形的底边。\\n";
    cin >> width;
    cout << " 三角形的面积是 " << height * width / 2 << "。\\n";

    return 0;
}
```

3.

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

int main()
{
    int sum=0, num=0;

    cout << " 请输入科目 1 的成绩。\\n";
    cin >> num;
    sum += num;
    cout << " 请输入科目 2 的成绩。\\n";
    cin >> num;
    sum += num;
    cout << " 请输入科目 3 的成绩。\\n";
    cin >> num;
    sum += num;
    cout << " 请输入科目 4 的成绩。\\n";
    cin >> num;
    sum += num;
    cout << " 请输入科目 5 的成绩。\\n";
    cin >> num;
    sum += num;
    cout << "5 个科目的总分是 " << sum << " 分。\\n";
    cout << "5 个科目的平均分是 " << (double)sum/5 << " 分。\\n";

    return 0;
}
```

另外，如果使用在第 6 章中学习的重复语句，则可以写出更加简单的代码。

## 第 5 章 具体情况具体处理

1.

Lesson

**A**

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

int main()
{
    int res;

    cout << " 请输入整数。\\n";
    cin >> res;

    if((res % 2)==0)
        cout << res << " 是偶数。\\n";
    else
        cout << res << " 是奇数。\\n";

    return 0;
}
```

因为非 0 的整数被判断为 true，所以也可以进行如下描述。

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

int main()
{
    int res;

    cout << " 请输入整数。\\n";
    cin >> res;

    if(res % 2)
        cout << res << " 是奇数。\\n";
    else
        cout << res << " 是偶数。\\n";
}
```

```
    return 0;
}
```

2.

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

int main()
{
    int num1, num2;

    cout << " 请输入两个整数。\\n";
    cin >> num1 >> num2;
    if (num1 < num2) {
        cout << num1 << " 比 " << num2 <<
            " 大。\\n";
    }
    else if(num1 > num2) {
        cout << num2 << " 比 " << num1 <<
            " 大。\\n";
    }
    else{
        cout << " 两个数为相同的值。\\n";
    }

    return 0;
}
```

3.

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

int main()
{
    int res;

    cout << " 请输入成绩。\\n";
    cin >> res;
```



```
cout << " 成绩为 " << res << "。 ";

switch(res){
case 1:
    cout << " 请再努力一些。 \n";
    break;
case 2:
    cout << " 再加油一下。 \n";
    break;
case 3:
    cout << " 还可以更好。 \n";
    break;
case 4:
    cout << " 做得很好。 \n";
    break;
case 5:
    cout << " 非常优秀。 \n";
    break;
}

return 0;
}
```

## 第 6 章 反复循环

1.

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

int main()
{
    cout << " 请输出 1~10 之间的偶数。 \n";
    for(int i=1; i<=10; i++){
        if((i % 2) == 0)
            cout << i << ' \n';
    }
}
```

```
    return 0;
}
```

2.

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

int main()
{
    int num, sum = 0;
    out << " 请输入考试的分数。(输入 0 时终止)\n";
    do{
        cin >> num;
        sum += num;
    }while(num);

    cout << " 考试总分为 " << sum << " 分。 \n";

    return 0;
}
```

3.

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

int main()
{
    for(int i=1; i<=5; i++){
        for(int j=0; j<i; j++){
            cout << '*';
        }
        cout << '\n';
    }

    return 0;
}
```

## 第 7 章 函数

1.

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

// square 函数的声明
int square(int x);

// square 函数的调用
int main()
{
    int num1;
    int sq1;

    cout << " 请输入整数。\\n";
    cin >> num1;
    sq1 = square(num1);
    cout << num1 << " 的平方是 " << sq1 << "。\\n";

    return 0;
}

// square 函数的定义
int square(int x)
{
    return x * x;
}
```

Lesson

**A**

2.

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

// square 函数的声明
int square(int x);
double square(double x);
```

```
// square 函数的调用
int main()
{
    int num1;
    int sq1;

    cout << " 请输入整数。\\n";
    cin >> num1;
    sq1 = square(num1);
    cout << num1 << " 的平方是 " << sq1 << "。\\n";

    double num2;
    double sq2;
    cout << " 请输入小数。\\n";
    cin >> num2;
    sq2 = square(num2);
    cout << num2 << " 的平方是 " << sq2 << "。\\n";

    return 0;
}

// square 函数 (int 型) 的定义
int square(int x)
{
    return x * x;
}

// square 函数 (double 型) 的定义
double square(double x)
{
    return x * x;
}
```

3.

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

// square 函数的定义
```

```
inline int square(int x){return x*x;}
inline double square(double x){return x*x;}

// square 函数的调用
int main()
{
    int num1;
    int sq1;
    cout << " 请输入整数。\\n";
    cin >> num1;
    sq1 = square(num1);
    cout << num1 << " 的平方是 " << sq1 << "。\\n";

    double num2;
    double sq2;
    cout << " 请输入小数。\\n";
    cin >> num2;
    sq2 = square(num2);
    cout << num2 << " 的平方是 " << sq2 << "。\\n";

    return 0;
}
```

4.

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

// squaret 函数模板的定义
template <class T>
T squaret(T x)
{
    return x * x;
}

// squaret 函数的调用
int main()
{
    int num1;
```

```

int sq1;
cout << " 请输入整数。\\n";
cin >> num1;
sq1 = squaret(num1);
cout << num1 << " 的平方是 " << sq1 << "。\\n";

double num2;
double sq2;
cout << " 请输入小数。\\n";
cin >> num2;
sq2 = squaret(num2);
cout << num2 << " 的平方是 " << sq2 << "。\\n";

return 0;
}

```

## 第8章 指针

1. ① ×      ② ○      ③ ×

2.

```

#include <iostream>
using namespace std;

// add 函数的声明
void add(int* x1, int* x2, int a);

int main()
{
    int num1 = 0;
    int num2 = 0;
    int ad = 0;

    cout << " 请输入两门科目的分数。\\n";
    cin >> num1 >> num2;
    cout << " 请输入加上的分数。\\n";
    cin >> ad;
    add(&num1, &num2, ad);
}

```

```
    cout << " 因为加了 "<<ad << " 分 \n";
    cout << " 科目一变成 " << num1 << " 分。 \n";
    cout << " 科目二变成 " << num2 << " 分。 \n";

    return 0;
}

// add 函数的定义
void add(int* x1, int* x2, int a)
{
    *x1 += a;
    *x2 += a;
}
```

3.

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

// add 函数的声明
void add(int& x1, int& x2, int a);

int main()
{
    int num1 = 0;
    int num2 = 0;
    int ad = 0;

    cout << " 请输入两门科目的分数。 \n";
    cin >> num1 >> num2;
    cout << " 请输入加上的分数。 \n";
    cin >> ad;
    add(num1, num2, ad);
    cout << " 因为加了 " <<ad << " 分 \n";
    cout << " 科目一变成 " << num1 << " 分。 \n";
    cout << " 科目二变成 " << num2 << " 分。 \n";

    return 0;
}
```

```
// add 函数的定义
void add(int& x1, int& x2, int a)
{
    x1 += a;
    x2 += a;
}
```

## 第9章 数组

1.

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

// max 函数的声明
int max(int x[]);

int main()
{
    int test[5];
    cout << " 请输入考试分数。\\n";
    for(int i=0; i<5; i++){
        cin >> test[i];
    }
    int m = max(test);
    cout << " 最高分是 " << m << " 分。\\n";

    return 0;
}

// max 函数的定义
int max(int x[])
{
    int m = x[0];
    for(int i=1; i<5; i++){
        if(m < x[i])
            m = x[i];
    }
}
```



```
}  
    return m;  
}
```

2.

```
# include <iostream>  
using namespace std;  
  
// length 函数的声明  
int length(char* str);  
  
int main()  
{  
    char str[100];  
    cout << " 请输入字符串。\\n";  
    cin >> str;  
    int ln = length(str);  
    cout << " 字符串的长度为 " << ln << "。\\n";  
  
    return 0;  
}  
  
// length 函数的定义  
int length(char* str)  
{  
    int i = 0;  
    while(str[i]){  
        i++;  
    }  
    return i;  
}
```

3.

```
# include <iostream>  
using namespace std;  
  
// count 函数的声明  
int count(char str[], char ch);
```

Lesson

**A**

```

int main()
{
    char str[100];
    char ch;

    cout << " 请输入字符串。\\n";
    cin >> str;
    cout << " 请输入要从字符串中搜索的字符。\\n";
    cin >> ch;
    int c = count(str, ch);
    cout << str << " 中有 " << c << " 个 " << ch << " 字符。\\n";

    return 0;
}

// count 函数的定义
int count(char str[], char ch)
{
    int i = 0;
    int c = 0;
    while(str[i]){
        if(str[i] == ch)
            c++;
        i++;
    }
    return c;
}

```

## 第 10 章 创建大型程序

1. ① ○    ② ○    ③ ×    ④ ×    ⑤ ○
2. 使用 delete 运算符释放内存。

```

# include <iostream>
using namespace std;

int main()

```

```
{  
    int* pA;  
    pA = new int;  
    *pA = 10;  
    delete pA;  
  
    return 0;  
}
```

## 第 11 章 各种类型

1.

```
# include <iostream>  
using namespace std;  
  
// 结构体 Person 的声明  
struct Person{  
    int age;  
    double weight;  
    double height;  
};  
  
int main()  
{  
    Person ps[2];  
  
    for(int i=0; i<2; i++){  
        cout << " 请输入年龄。\\n";  
        cin >> ps[i].age;  
        cout << " 请输入体重。\\n";  
        cin >> ps[i].weight;  
        cout << " 请输入身高。\\n";  
        cin >> ps[i].height;  
    }  
  
    for(int j=0; j<2; j++){  
        cout << " 年龄 " << ps[j].age << " 体重 " << ps[j].weight
```

```
        << " 身高 " << ps[j].height << "。 \n";  
    }  
    return 0;  
}
```

2.

```
# include <iostream>  
using namespace std;  
  
// 结构体型 Person 的声明  
struct Person{  
    int age;  
    double weight;  
    double height;  
};  
  
// aging 函数的定义  
void aging(Person* p)  
{  
    p->age++;  
}  
  
int main()  
{  
    Person ps;  
  
    cout << " 请输入年龄。 \n";  
    cin >> ps.age;  
    cout << " 请输入体重。 \n";  
    cin >> ps.weight;  
    cout << " 请输入身高。 \n";  
    cin >> ps.height;  
  
    cout << " 年龄 " << ps.age << " 体重 " << ps.weight << " 身高 "  
        << ps.height << "。 \n";  
  
    aging(&ps);  
    cout << " 一年后。 \n";
```

```

    cout << "年龄" << ps.age << "体重" << ps.weight << "身高"
        << ps.height << "。\\n";

    return 0;
}

```

## 第 12 章 类的基本

1. ① ×    ② ○    ③ ×    ④ ○    ⑤ ○  
 2.

```

#include <iostream>
using namespace std;

// Point 类的声明
class Point{
private:
    int x;
    int y;
public:
    void setX(int a);
    void setY(int b);
    int getX(){return x;}
    int getY(){return y;}
};

// Point 类成员函数的定义
void Point::setX(int a)
{
    if(a >= 0 && a <= 10)
        x = a;
    else
        x = 0;
}

void Point::setY(int b)
{
    if(b >= 0 && b <= 10)

```

```
        y = b;
    else
        y = 0;
}

int main()
{
    Point p;
    int x, y;

    cout << " 请输入 X 坐标。\\n";
    cin >> x;
    cout << " 请输入 Y 坐标。\\n";
    cin >> y;

    p.setX(x);
    p.setY(y);

    cout << " 坐标是 (" << p.getX() << ", " << p.getY() << ")。\\n";

    return 0;
}
```

## 第13章 类的功能

1. ① ×      ② ○      ③ ×
2. ① ×      ② ×      ③ ×

## 第14章 新类型

1. ① ×      ② ×      ③ ○
2. ① ×      ② ○      ③ ○
3. main() 函数内的 drv.showBs (); 的调用很模糊。

## 第15章 关于类的高级论题

1. ① ○      ② ○      ③ ×

2. ① ×      ② ○      ③ ×

3.

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

// Point 类的声明
class Point{
    private:
        int x;
        int y;
    public:
        Point(int a=0, int b=0){x=a; y=b;}
        void show(){cout<< "x:" << x << " y:" << y <<'\n';}
        Point operator++();
        Point operator++(int d);
        Point operator--();
        Point operator--(int d);
        friend Point operator+(Point p1, Point p2);
        friend Point operator+(Point p, int a);
        friend Point operator+(int a, Point p);
        friend Point operator-(Point p1, Point p2);
        friend Point operator-(Point p, int a);
        friend Point operator-(int a, Point p);
};

// Point 类成员函数的定义
Point Point::operator++()
{
    x++;
    y++;
    return *this;
}
Point Point::operator++(int d)
{
    Point p = *this;
    x++;
    y++;
```

Lesson

**A**

```
        return p;
    }
    Point Point::operator--()
    {
        x--;
        y--;
        return *this;
    }
    Point Point::operator--(int d)
    {
        Point p = *this;
        x--;
        y--;
        return p;
    }

    // 友元函数的定义
    Point operator+(Point p1, Point p2)
    {
        Point tmp;
        tmp.x = p1.x + p2.x;
        tmp.y = p1.y + p2.y;
        return tmp;
    }
    Point operator+(Point p, int a)
    {
        Point tmp;
        tmp.x = p.x + a;
        tmp.y = p.y + a;
        return tmp;
    }
    Point operator+(int a, Point p)
    {
        Point tmp;
        tmp.x = a + p.x;
        tmp.y = a + p.y;
        return tmp;
    }
}
```



```
Point operator-(Point p1, Point p2)
{
    Point tmp;
    tmp.x = p1.x - p2.x;
    tmp.y = p1.y - p2.y;
    return tmp;
}

Point operator-(Point p, int a)
{
    Point tmp;
    tmp.x = p.x - a;
    tmp.y = p.y - a;
    return tmp;
}

Point operator-(int a, Point p)
{
    Point tmp;
    tmp.x = a - p.x;
    tmp.y = a - p.y;
    return tmp;
}

int main()
{
    Point p1(20,10);
    Point p2(1,2);
    p1 = p1-p2;
    p1--;
    p1 = p1-3;
    p2 = 3-p2;

    p1.show();
    p2.show();
    return 0;
}
```

4. 同第 3 题答案。

## 第16章 文件和流

1.

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

int main()
{
    for (int i=0; i<=5; i++){
        for (int j=1; j<=5; j++){
            cout.width(3);
            cout.fill('-');
            cout << i*5+j;
        }
        cout << '\n';
    }
    return 0;
}
```

2.

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

int main(int argc, char* argv[])
{
    if(argc != 2){
        cout << " 参数的个数不同。\\n";
        return 1;
    }

    ifstream fin(argv[1]);
    if(!fin){
        cout << " 打不开文件。\\n";
        return 1;
    }
}
```

```
}

const int num = 8;
int test[num];
for(int i=0; i<num; i++){
    fin >> test[i];
}
int max = test[0];
int min = test[0];
for(int j=0; j<num; j++){
    if(max < test[j])
        max = test[j];
    if(min > test[j])
        min = test[j];
    cout << "No." << j+1 << setw(5) << test[j] << '\n';
}

cout << "最高分是 " << max << "。 \n";
cout << "最低分是 " << min << "。 \n";

fin.close();

return 0;
}
```

