1-2

面向对象的语言的设计出发点是为了能更直接的描述客观世界中存在的事物（对象）以及它们之间的关系。通过继承与多态方便的实现了代码重用，大大缩短了软件开发周期，并使软件风格统一，能比较直接地反应问题域的本来面目。编程人员能运用人类认知事物的一般思维方法进行软件开发。

1-4

对象：系统中用来描述客观事物的一个实体，它是用来构成系统的一个基本单位。对象由一组属性和一组行为构成。

面向对象的方法：将数据及对数据的操作方法封装在一起，作为一个相互依存、不可分离的整体——对象，对同类型对象抽象出其共性，形成类。类中的大多数数据只能用本类的方法进行处理。类通过一个简单的外部接口，与外界发生关系，对象与对象之间通过消息进行通信。

优点：（1）程序模块间的关系更为简单，程序模块的独立性、数据的安全性就有了良好的保障。（2）通过继承与多态性，可以大大提高程序的可重用性，使得软件的开发和维护都更为方便。

1-9 （5） 二进制：0000 0000 0000 0000 1111 1111 1111 1111 十六进制：0000 FFFF

正数的补码为其本身，在32位机器中，有65535=0000 0000 0000 0000 1111 1111 1111 1111

其十六进制补码为 0000 FFFF

（6）二进制：11111111 十六进制：FF

-1的原码为 1000 0000 0000 0000 0000 0000 0000 0001

-1的反码为 1111 1111 1111 1111 1111 1111 1111 1110

-1的补码为 1111 1111 1111 1111 1111 1111 1111 1111

其十六进制补码为 FFFF FFFF

1-10 （2）143

原式=1×2^0+1×2^1+1×2^2+1×2^3+1×2^7=143

（5）11582

原式=2×16^3+13×16^2+3×16^1+14×16^0=11582

1-11

原码是一种计算机中对数字的二进制定点表示方法。原码表示法在数值前面增加了一位符号位（即最高位为符号位）：正数该位为0，负数该位为1，其余位表示数值的大小。

反码是数值存储的一种，但是由于补码更能有效表现数字在计算机中的形式，所以多数计算机一般都不采用反码表示数。正数的反码与原码表示相同，负数的反码与原码有如下关系。反码只是求补码的中间码。

计算机中，负数的补码由该数反码的末位加 1 求得，对补码再求补即得到原码。

2-1

优点

（1）C++解决了C语言中存在的一些问题

（2）C++支持面向面向对象的程序设计，引入了“类”的机制，满足了运用面向对象方法开发软件的需要

特点

（1）尽量兼容C，许多C代码不经修改就可以在C++中运行

（2）支持面向对象的特征，

2-5 BLUE=105

2-8 （1）50 （2）1 （3）50.25

2-11 n=100

2-23 (1) 1 (2) 0 (3) 0

2-24 (1)-1 (2) 1 (3) 1 (4) 3

2-25 (1) 1 (2) -1 (3) 0 (4) 0

2-27

#include<iostream>

using namespace std;

int main() {

int a;

cout << "你考试考了多少分？(0~100)";

cin >> a;

if ((a >= 0) && (a < 60)) cout << "差" << endl;

else if ((a >= 60) && (a < 80)) cout << "中" << endl;

else if ((a >= 80) && (a < 90)) cout << "良" << endl;

else cout << "优" << endl;

return 0;

}

2-29

（1）

#include<iostream>

using namespace std;

int main() {

int j;

for (int i = 2; i <= 100; i++)

{

for (j = 2; j <= i / 2; j++)

{

if (i%j == 0) break;

}

if (j > i / 2)

cout << i << " ";

}

cout << endl;

}

（2）

#include<iostream>

using namespace std;

int main() {

int i = 2, j=2 ;

while (i <= 100)

{

j = 2;

while (j<= i / 2) {

{if (i%j != 0) j++;

else break;

}

if (j > i / 2) cout << i<<" ";

}

i++;

}

}

（3）

#include<iostream>

using namespace std;

int main() {

int i = 2, j=2 ;

do

{

j = 2;

do{

{if (i%j != 0) j++;

else break;

}

if (j > i / 2) cout << i<<" ";

} while (j <= i / 2);

i++;

} while (i <= 100);

}

2-32

（1）

#include<iostream>

using namespace std;

int main() {

int n = 1 + rand() % 100, a;

cout << "请输入一个1-100的数字" << endl;

cin >> a;

for (a; a != n; cin >> a)

{

if (a > n) cout << "请输入一个更小的数字";

else cout << "请输入一个更大的数字";

}

cout << "恭喜您，猜对了！" << endl;

return 0;

}

（2）

#include<iostream>

using namespace std;

int main() {

int n = 1 + rand() % 100, a;

cout << "请输入一个1-100的数字" << endl;

cin >> a;

while (a > n) {

cout << "请输入一个更小的数字" << endl;

cin >> a;

};

while (a < n) {

cout << "请输入一个更大的数字" << endl;

cin >> a;

};

if (a = n) cout << "恭喜您！猜对了！" << endl;

return 0;

}

（3）

#include<iostream>

using namespace std;

int main() {

int n = 1 + rand() % 100, a;

cout << "请输入一个1-100的数字" << endl;

cin >> a;

do {

cout << "请输入一个更小的数字" << endl;

cin >> a;

} while (a > n);

do {

cout << "请输入一个更大的数字" << endl;

cin >> a;

} while (a < n);

if (a = n) cout << "恭喜您！猜对了！" << endl;

return 0;

}

2-35

#include<iostream>

using namespace std;

int main() {

int i = 1, j = 1;

for (i = 1; i <= 9; i++)

{

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

{

cout << i << "\*" << j << "=" << i\*j << " ";

}

cout << endl;

}

return 0;

}