1-2.面对对象的编程优点在于，它可以通过类的继承与多态可以很方便地实现代码重用，大大缩短了软件开发周期，并使得软件风格统一。它使程序能够比较直接地反映问题域的本来面目，软件开发人员能够利用人类认识事物所采用的一般思维方法来进行软件开发。

1-4.一般意义上的对象是现实世界中一个实际存在的事物。在面向对象方法中的对象，是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。

面向对象方法是指：首先它将数据及对数据的操作方法放在一起，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。

它的特点在于，它使得程序模块间的关系更为简单，程序模块的独立性、数据的安全性有了良好的保障；同时，它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）65535=（1111111111111111）2=（FFFF）4

（6）-1=（11111111）2=（FF）4

1-11.对于正数而言，原码反码与补码相同；

对于负数而言，反码的符号位与原码相同，仍用1表示，其余各位取反（0变1，1变0）。

负数的补码由该数反码的最末位加1求得。

2-1.c++语言的主要特点表现在两个方面，一是尽量兼容C，二是支持面向对象的方法。

它的缺点在于，它的语言本身过度复杂，人类难于理解其语义。并且由于受到C++复杂性的影响，非常难于编写。

2-5.BLUE的值为200

2-8.（1）50 （2）1 （3）50.0

2-11.n=100

2-23.（1）true （2）false （3）true

2-24.（1）-1 （2）3 （3）3 （4）3

2-25 （1）1 （2）-127 （3）1 （4）0

2-27

#include<iostream>

using namespace std;

int main(){

int n;

cout<<"How many scores did you get in the exam?(0~100): ";

cin>>n;

cout<<"Your level is : ";

if(n>=90&n<=100)

cout<<"Excellent"<<endl;

else if(n>=80&n<90)

cout<<"Great"<<endl;

else if(n>=60&n<80)

cout<<"Good"<<endl;

else if(n>=0&n<60)

cout<<"Bad"<<endl;

else

cout<<"Your score is out of the range 0~100"<<endl;

return 0;

}

2-29

#include<iostream>

using namespace std;

int main(){

int i,j;

bool flag;

cout<<"The prime numbers between 1~100 is : ";

i=2;

while(i<=100){

j=2;

flag=true;

while(j<i){

if(i%j==0){

flag=false;

break;}

j++;

}

if(flag) cout<<i<<" ";

i++;

}

return 0;

}

2-35.

#include<iostream>

using namespace std;

int main()

{

int a,b;

a=1,b=1;

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

{

for(b=1;b<=a;b++)

{

cout<<a\*b<<" ";

}

cout<<endl;

}

return 0;

}

2-32#include <iostream>

#include <time.h>

using namespace std;

int main()

{

int num,a;

srand(time(0));

num=rand()%100+1; //1~100随机生成一个数

cout<<num<<endl;

while(num)

{

cout<<"count in a number ：";

cin>>a;

if(a>num)

{

cout<<"more than the number "<<endl;

}

else if(a<num)

{

cout<<"less than the number "<<endl;

}

else

{

num=num-a;

cout<<"succeed！"<<endl;

}

}