1.

（a）上述源文件的名字是什么？ Person.java

（b）编译上述源文件将生成几个字节码文件？这些字节码文件的名字都是什么？

将生成两个字节码文件，名字为Person.class，Xiti.class。

（c）在命令行执行 java Person 得到怎样的错误提示？执行 java xiti 得到怎样的错误提示？执行 java Xiti.class 得到怎样的错误提示？执行 java Xiti 得到怎样的输出结果?

执行 java Person

```
C:\java>java Person
错误: 在类 Person 中找不到 main 方法, 请将 main 方法定义为:
   public static void main(String[] args)
否则 JavaFX 应用程序类必须扩展javafx.application.Application
```

执行 java xiti

```
C:\java>java xiti
错误: 找不到或无法加载主类 xiti
```

执行 java Xiti.class

```
C:\java>java Xiti.class
错误: 找不到或无法加载主类 Xiti.class
```

执行 java Xiti

```
C:\java>java Xiti
您好，很高兴认识您 nice to meet you
```

2.运行截图

```
C:\java>javac Cxy.java

C:\java>java Cxy
name:Chenxinyu
student id:18373540
```

3.

（1）运行截图

```
Hello World!
a is 4
a is 11
```

（2）运行截图

Name	Value
no method return value	
args	String[0] (id=16)
a	2

4.运行结果

```
b+c=1.8
b+c=0.21.6
字符串不是基本数据类型
Hello World
```

5.运行结果

```
字节型变量 b = 85
短整型变量 s = 22015
整型变量 i = 1000000
长整型变量 l = 65535
字符型变量 c = c
浮点型变量 f = 0.23
双精度变量 d = 7.0E-4
布尔型变量 B = true
字符串对象 S = This is a string
```

6.运行结果

基本类型	默认值	基本类型	默认值
byte	0	**boolean**	false
short	0	**char**	\u0000

int	0	**float**	0.0
long	0	**double**	0.0

7.运行结果

```
'大'的Unicode编码：22823
Unicode编码为23398的字符是：学
```

8.运行结果

```
'你'的Unicode编码：20320
'我'的Unicode编码：25105
'他'的Unicode编码：20182
```

9.运行结果

```
'a'的十六进制位的Unicode编码是\u0061
'冲'的十六进制位的Unicode编码是\u51b2
```

10.运行结果

```
max=96.9
min=-9.9
```

11.运行结果

```
1
2
b is false
```

请解释 Java 执行串联逻辑运算时的流程，可以用文字、流程图、伪代码描述。

在 java 中逻辑运算符有“&&”、“||”和“！”，分别表示“且”、“或”和“非”。其中“&&”和“||”是二目运算符，“！”是单目运算符。首先考虑逻辑运算符两边的符号（若有）的优先级，从左到右， 依次进行计算。 其中逻辑运算符&&”和“||”也称作短路逻辑运算符，这是因为对于逻辑运算 op1&&op2 中，当 op1 的值为 false 时，“&&”运算符在进行运算时不再去计算 op2 的值，就可以直接得出 op1&&op2 的结果为 false；对于逻辑运算 op1||op2 中，当 op1 的值为 true 时，“||”运算符在进行运算时不再去计算 op2 的值，就可以直接得出 op1&&op2 的结果为 true.

思考如何利用短路这个机制来优化程序。

可以减少 if 语句的条件判断，例如：

```
if(a){
    if(b){
        if(c){
        ......
        }
    }
}
```

可以优化为：

```
if(a&&b&&c){
    ......
}
```

也可以减少 for 语句的条件判断，例如：

```
for(...;a;...){
    for(...;b;...){
        for(...;c;...){
        ......
        }
    }
}
```

可以优化为：

```
for(...;a&&b&&c;...){
    ......
}
```

12.运行结果

```
全局变量 a = 10
在print()中, 全局变量 a = 10, b = 20.0
在print()中, 全局变量 a = 30, b = 20.0
全局变量变化后 a = 30
```

13.运行结果

```
在print()中的局部变量 a = 10, b = 20.0
在print()中的局部变量 a = 30, b = 20.0
```

14.运行结果

```
intArray[0]=8
intArray[1]=9
intArray[2]=12
intArray[3]=13
intArray[4]=14

sum=56
```

15.运行结果

```
b[0][0]=1000
sum=1139
b.lenght=3
 arr1:
 0 1 2 3 4 5 6 7 8 9 10 11
 arr2:
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 arr3:
 0 1 2 3 4 5 6 7 8
```

16.

运行结果

```
Jeep好好
```

原因:

```
for(int i=1;i<=4;i++) {
    switch(i) {
    case 1: c = 'J';
    System.out.print(c);
    case 2: c = 'e';
    System.out.print(c);
    break;
    case 3: c = 'p';
    System.out.print(c);
    default: System.out.print("好");
    }
}
```

当 i=1 时，执行 case 1，输出‘J’，但是 case 1 执行完之后并没有 break 语句，所以继续执行下一个情况 case 2，输出‘e’，在 case 2 情况中有 break 语句，所以跳出开关语句；当 i=2 时，执行 case 2，输出‘e’，在 case 2 情况中有 break 语句，所以跳出开关语句；当 i=3 时，执行 case 3，输出‘p’，但是 case 3 执行完之后并没有 break 语句，所以继续执行下一个情况 default，输出‘好’，跳出开关语句；当 i=4 时，没有这种情况，所以执行 default 语句，输出‘好’，跳出开关语句。所以，最终输出结果为“Jeep 好好”。

17.运行结果

```
do-while循环计算结果为：1.7182818284590455
for循环计算结果为：1.7182818284590455
```

18.运行结果

```
5
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
```

19.运行结果

```
5
  *
 ***
*****
 ***
  *
```

20.运行结果

```
50000
3.141612653189785
```

21.运行结果

```
12 34 3 5 2 11
2 3 5 11 12 34
```

22.运行结果

```
-45 12 45 67 67 89 123
45
45在数组中的第3位
```

23.运行结果

```
A n   A f r i c a n   S w a l l o w
```

24.运行结果

```
p1的引用:fifth.Point@15db9742
p2的引用:fifth.Point@6d06d69c
p1的x,y坐标:1111,2222
p2的x,y坐标:-100,-200
p1的引用:fifth.Point@6d06d69c
p2的引用:fifth.Point@6d06d69c
p1的x,y坐标:0,0
p2的x,y坐标:0,0
```

注释标注出的四行输出的内容是：

p1 的 x,y 坐标:1111,2222

p2 的 x,y 坐标:-100,-200

p1 的 x,y 坐标:0,0

p2 的 x,y 坐标:0,0

25.

运行结果

```
数组a的元素个数=4
数组b的元素个数=3
数组a的引用=[I@15db9742
数组b的引用=[I@6d06d69c
数组a的元素个数=3
数组b的元素个数=3
a[0]=100,a[1]=200,a[2]=300
b[0]=100,b[1]=200,b[2]=300
```

原因:

首先构造了两个数组 a，b 并初始化，根据初始化的元素个数，数组 a 的元素个数为 4，分别是 1，2，3，4；数组 b 的元素个数为 3，分别是 100，200，300；并且为两个数组分配了不同的内存地址，其中数组 a 的引用为[I@15db9742，数组 b 的引用为[I@6d06d69c，接下来将 b 的引用赋值给 a,此时 a 的引用不再为[I@15db9742，而是[I@6d06d69c，即 a 和 b 指向同一块内存地址，这块内存地址顺序存储着三个连续的数组元素 100，200，300，所以最终数组 a 的元素个数为 3，分别是 100，200，300。

26.运行结果

```
1
2
-1
-2
-3
-4
9
7
6
```

27.运行结果

```
cxyxxn
cccxxx
```

28.运行结果

```
小王原本的体重是70
小王减肥后的体重是45
```

29.运行结果

```
mom买meat剩下17钱
aunt买vegetables剩下5钱
cxy买eggs and condiments剩下28钱
dad做cooking
aunt1做dessert
```