**四川大学期末考试答案（闭卷）**

**（2016——2017学年第 1 学期） B卷**

课程号： 304007030 课序号： 课程名称：JAVA程序设计

**一，（本大题30分，其中每题1分）**

1．BCBBB 6．CCCBD 11.CBDBC 16.BADDB 21.BDAAA 26.DCDCB

二，**（本大题15分，其中每题3分）**

DDCAB

**三、填空题（本大题20分，其中每个空格2分）**

1.  main

2．属性,行为

3． 类名, 默认构造

4．局部变量，初始化  
5. 字节码，.class

6. 2

四，**阅读程序，写出程序的输出结果（本大题20分，其中每题5分）**

1, condition 1

Finally

2, 10

3, south   east to west

4,the value of a[0] is 0the value is a[5] is 5

**五，编程题（本大题25分）**

1，

class Course {

private String courseName;

private String[] students = new String[100];

private int numberOfStudents;

//1分

public Course(String courseName) {

this.courseName = courseName;

}

//4分

public void addStudent(String student) {

students[numberOfStudents] = student;

numberOfStudents++;

}

//1分

public String[] getStudents() {

return students;

}

//1分

public int getNumberOfStudents() {

return numberOfStudents;

}

//1分

public String getCourseName() {

return courseName;

}

//3分

public void dropStudent(String student) {

int i;

for(i=0;i< numberOfStudents;i++)

{

if (students[i].equals(student))

break;

}

if (i< numberOfStudents) {

for(int j= i;j<numberOfStudents-1;j++)

students[j]=students[j+1];

students[numberOfStudents-1]=null;

numberOfStudents--;

}

}

}

public class TestCourse {//4分

public static void main(String[] args) {

Course course1 = new Course("Data Structures");

Course course2 = new Course("Database Systems");

course1.addStudent("Peter Jones");

course1.addStudent("Brian Smith");

course1.addStudent("Anne Kennedy");

course2.addStudent("Peter Jones");

course2.addStudent("Steve Smith");

System.out.println("Number of students in course1: "

+ course1.getNumberOfStudents());

String[] students = course1.getStudents();

for (int i = 0; i < course1.getNumberOfStudents(); i++)

System.out.print(students[i] + ", ");

System.out.println();

System.out.print("Number of students in course2: "

+ course2.getNumberOfStudents());

}

}

2. class StackOfIntegers {

//写出属性2分

private int[] elements;

private int size;

public static final int DEFAULT\_CAPACITY = 16;

/\*\* Construct a stack with the default capacity 16 \*/

public StackOfIntegers() {

this(DEFAULT\_CAPACITY);

}

/\*\* Construct a stack with the specified maximum capacity \*/

public StackOfIntegers(int capacity) {

elements = new int[capacity];

}

/\*\* Push a new integer into the top of the stack \*/

//2分

public void push(int value) {

if (size >= elements.length) {

int[] temp = new int[elements.length \* 2];

System.arraycopy(elements, 0, temp, 0, elements.length);

elements = temp;

}

elements[size++] = value;

}

/\*\* Return and remove the top element from the stack \*/

//2分

public int pop() {

return elements[--size];

}

/\*\* Return the top element from the stack \*/

public int peek() {

return elements[size - 1];

}

/\*\* Test whether the stack is empty \*/

//2分

public boolean empty() {

return size == 0;

}

/\*\* Return the number of elements in the stack \*/

public int getSize() {

return size;

}

}

public class TestStackOfIntegers {//2分

public static void main(String[] args) {

StackOfIntegers stack = new StackOfIntegers();

for (int i = 0; i < 10; i++)

stack.push(i);

while (!stack.empty())

System.out.print(stack.pop() + " ");

}

}