

# 考试科目名称 计算与软件工程 I (A 卷)

考试方式: 闭卷 考试日期\_\_\_\_年\_\_\_\_月\_\_\_\_日 教师\_\_\_\_\_

系(专业)\_\_\_\_\_ 年级\_\_\_\_\_ 班级\_\_\_\_\_

学号\_\_\_\_\_ 姓名\_\_\_\_\_ 成绩\_\_\_\_\_

题号	一	二	三	四	五	六	七	八	九	十
分数										

得分 一、 选择题 (本题满分 20 分)

- 下列标识符合法的是[ ]  
A. new B. \$Usdollars C. 1234 D. car.taxi
- 下列说法正确的是[ ]  
A. class 中的 constructor 不可省略  
B. constructor 必须与 class 同名, 但方法不能与 class 同名  
C. constructor 在一个对象被 new 时执行  
D. 一个 class 只能定义一个 constructor
- 下面哪个陈述是正确的?[ ]  
A. 接口编译成独立的字节码文件  
B. 抽象类可以有使用该抽象类构造方法创建的实例  
C. 非抽象父类的子类不能是抽象类  
D. 子类不能覆盖父类中的具体方法, 将其声明成抽象方法
- 若类 A 是类 B 的子类, 下列说法错误的是[ ]  
A. new A[10] instanceof B[ ]  
B. new B[10] instanceof Object[ ]  
C. new B[10] instanceof Object  
D. 上述说法都不正确
- 下面关于在继承中访问控制权限说法错误的是[ ]  
A. 在子类内部能访问 public 和 protected 继承成员, 不能访问 private 继承成员  
B. 使用子类的程序能访问子类的 public 继承成员, 但不能访问 protected 和 private 继承成员  
C. 在子类内部能访问子类自己定义的所有成员  
D. 使用子类的程序能访问子类自己定义的 public 和 protected 成员, 但不能访问子类自己定义的 private 继承成员
- 类 Parent、Child 分别定义如下:

```

1. public class Parent {
2.     public float aFun(float a, float b) throws IOException {    }
3. }
4. public class Child extends Parent{
5.
6. }

```

将以下哪种方法插入第 5 行是不合法的? [                      ]

- A. float aFun(float a, float b){ }
- B. public int aFun(int a, int b) throws Exception{ }
- C. public float aFun(float p, float q){ }
- D. public int aFun(int a, int b) throws IOException{ }

7. 请阅读程序片段:

```

public class Example{
    String str=new String("good");
    char ch[]={ 'a','b','c'};
    public static void main(String args[]){
        Example ex=new Example();
        ex.change(ex.str,ex.ch);
        System.out.println(ex.str+" and "+ex.ch[0]+ex.ch[1]+ex.ch[2]);
    }
    public void change(String str,char ch[]){
        str="test ok";
        ch[0]='g' ;
    }
}

```

该程序片段的输出结果是: [                      ]

- A. good and abc
- B. good and gbc
- C. test ok and abc
- D. test ok and gbc

8. 请阅读程序片段:

```

1. public class NewGarb {
2.     public static Object getIt() {
3.         Object rg = new Integer(3);
4.         Object dg[][] = new Object[1][2];
5.         dg[0][1] = rg;
6.         dg[0][0] = rg;
7.         rg = null;
8.         return rg;
9.     }
10. }

```

下列哪条语句是正确的? [                      ]

- A. 类NewGarb不能正确编译
- B. 方法getIt()一定不能是static的
- C. 类 NewGarb 能正确编译，但会发生异常因为对象 dg 没有初始化为 null
- D. 当 getIt()方法调用返回后，垃圾收集器可以合法地回收对象 rg

9. 请阅读程序片段：

```
public class Outer{
    public String name = "Outer";
    public static void main(String args[]){
        Inner i = new Inner();
        i.showName();
    }
    private class Inner{
        String name =new String("Inner");
        void showName(){ System.out.println(name);}
    }
}
```

当试图编译并运行程序时，下面那个语句是正确的？ [                      ]

- A. 程序正确，输出为 “Outer”
  - B. 程序正确，输出为 “Inner”
  - C. 程序不正确，因为 Inner 为 private 变量
  - D. 程序不正确，因为在 main（）中建立的 Inner 对象必须与类 Outer 的对象相关
10. 下面有关 java 代码安全性的叙述哪个是错误的？ [                      ]
- A. 运行时解释器执行代码
  - B. 字节码校验器加载查询执行需要的所有类
  - C. 在运行时，字节码被加载，验证后在解释器里面运行
  - D. 类加载器通过分离本机文件系统的类和从网络导入的类增加安全性

得分	
----	--

二、 简答题（本题满分 20 分）

1. 请分别简述面向对象程序设计和面向过程程序设计的特点。

2. 什么叫多态？ Java 中实现方法多态的两种机制及其特点是什么？

3. 简述面向对象中类之间的三种关系，并说明在 **Java** 中是如何实现的？

4. 什么是包？简述在 **Java** 中包的作用。

得分	
----	--

 三、阅读程序，并写出结果（本题满分 20 分）

```
(1) class Test {  
    static int value = 1;  
    static int f(int n) {  
        int temp;  
        if (n == 0 || n == 1)  
            return n;  
        else { temp = f(n - 1) + f(n - 2);  
            if (temp > value) {  
                System.out.println(value + "");  
                Value = temp; }  
            return temp; }  
    }  
    public static void main(String a[]) {  
        int n = 7;  
        System.out.println("0 1");  
        f(n);  
        System.out.println("\n");  
    }  
}
```

输出结果: \_\_\_\_\_

(2) class SuperShow {

```

    public String str = "SuperStr";
    public void show() {
        System.out.println("Super.show:" + str); }
}
class ExtendShow extends SuperShow {
    public String str = "ExtendStr";
    public void show() {
        system.out.println("Extend.show" + str);    }
    public static void main(String[] args) {
        ExtendShow ext = new ExtendShow();
        SuperShow sup = ext;
        system.out.println("sup.str="+sup.str);
        system.out.println("ext.str =" +ext.str);
        sup.show();
        ext.show();    }
}

```

输出结果: \_\_\_\_\_

```

(3) import java.io.*;
public class Exercise {
    public static void main(String[] args)throws IOException {
        BufferedReader input =
            new BufferedReader(new FileReader(args[1]));
        BufferedWriter output =
            new BufferedWriter(new FileWriter("temp111.txt"));
        String line;
        while ((line = input.readLine()) != null) {
            String s = line.replaceAll(args[0], "");
            output.write(s+ "\n");        }
        input.close();
        output.close();
        input = new BufferedReader(new FileReader("temp.txt"));
        output = new BufferedWriter(new FileWriter("Exercise.txt"));
        while ((line = input.readLine()) != null) {
            output.write(line + "\n");    }
        input.close();
        output.close();    }
}

```

程序实现的功能: \_\_\_\_\_

```

(4)
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class KeyExample extends JFrame {
    private KeyboardPanel keyboardPanel = new KeyboardPanel();
}

```

```

        public KeyExample() { add(keyboardPanel);
                               keyboardPanel.setFocusable(true); }

    public static void main(String[] args) {
        KeyExample frame = new KeyExample();
        frame.setTitle("KeyExample");
        frame.setLocationRelativeTo(null);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 300);
        frame.setVisible(true);    }

    static class KeyboardPanel extends JPanel {
        private int x = 100;
        private int y = 100;
        private char keyChar = 'A';
        public KeyboardPanel() {
            addKeyListener(new KeyAdapter() {
                public void keyPressed(KeyEvent e) {
                    switch (e.getKeyCode()) {
                        case KeyEvent.VK_DOWN: y += 10; break;
                        case KeyEvent.VK_UP: y -= 10; break;
                        case KeyEvent.VK_LEFT: x -= 10; break;
                        case KeyEvent.VK_RIGHT: x += 10; break;
                        default: keyChar = e.getKeyChar();    }
                    repaint();    }
            });
        }

        protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            g.setFont(new Font("TimesRoman", Font.PLAIN, 24));
            g.drawString(String.valueOf(keyChar), x, y);    }
    }
}

```

程序实现的功能： \_\_\_\_\_

得分	
----	--

四、阅读程序，回答问题（共 12 分）

```

1. public class Example {
2.     public void method1( int M) {
3.         try { System.out.println("Entering try block");
4.             method2(M);
5.             System.out.println("Exiting try block");
6.         } catch (Exception e) { System.out.println("Error:" + e.getMessage());}
7.     }

```

```

8. public void method2( int M) {
9.     if (M > 100) throw new ArithmeticException(M + "is too large"); }
10. public static void main(String args[ ]) {
11.     Example ex = new Example();
12.     ex.method1(500); }
13. }

```

(1) 上述 **Example** 程序正在执行第 9 条语句，请画出表现这种情形的方法调用栈示意图；

(2) **Example** 程序运行时，它会打印什么内容？

(3) 如果 **Example** 程序的 **main()**方法中的第 12 句变成 **ex.method1(5)**，它会打印什么内容？

(4) 如果 **Example** 程序抛出的异常是 **Exception** 而不是 **ArithmeticException**，解释为什么会得到出错消息：“**java.lang.Exception must be caught, or it must be declared...**”

得分	
----	--

五、指出下列程序中划线部分是否正确，如果正确，说明该语句的功能；如果不正确，请说明原因（16 分）

```

class Point {
    private int x,y;
    Point(int x, int y){
        this.x=x; this.y=y;    }
    Point() { this (0, 0); }      (1)_____
    Point get() { return this; }  (2)_____
    protected void put(int x, int y) { this.x=x; this.y=y; }
}
class Point3D extends Point {
    int z;
    Point3D(int x, int y, int z){
        this.x=x; this.y=y; this.z=z; (3)_____
    }
}

```

```

void put(int x, int y, int z){
    super.put (x,y);    this.z=z;      (4)_____
}
public static void main(String args[]){
    Point3D p = new Point3D(1,1,1);
    p.put(10,10); p.put ( 5, 5, 5);    (5)_____
    Point p1=new Point(15,15);
    System.out.println(p1.x);          (6)_____
    Point p2 = new Point3D (8, 8,8);    (7)_____
    p2.get();                          (8)_____
}
}

```

得分	
----	--

六、按要求实现下列程序（12分）：

自己设计一个日期类，实现 **Comparable** 接口，可以输入年月日作为构造函数的参数，如果不使用参数，则设定为 **1900** 年 **1** 月 **1** 日；编写一个方法 **equals**（）判断两个日期是否相等；实现 **Comparable** 接口中的 **compareTo**（）方法进行日期比较，返回两个日期之间的相差天数。