

SINGLE CHOICE.

Identify the choice that *best* completes the statement or answers the question.

The following is 1 point for each question.

_____ 1. Given the following code:

```
class Test{  
    private int m;  
    public static void fun() {  
        // some code...  
    }  
}
```

How can the member variable m be accessible directly in the method fun()?

- A. change private int m to protected int m
- B. change private int m to public int m
- C. change private int m to static int m
- D. change private int m to int m

_____ 2. Given the following expression: `int m[] = {0, 1, 2, 3, 4, 5, 6 };`

Which result of the following expressions equals to the number of the array elements?

- A. `m.length()`
- B. `m.length`
- C. `m.length()+1`
- D. `m.length+1`

_____ 3. Which keyword is used to allow interaction with the lock flag?

- A. native
- B. static
- C. synchronized
- D. abstract

_____ 4. Given the following fragment of code:

```
Double d1 = new Double(1.0);  
Double d2 = new Double(1.0);  
Float f = new Float(1.0F);
```

Which results of the following expressions are true?

- A. `d1 == d2`
- B. `d1.equals(d2)`
- C. `d1 = f`
- D. `f.equals(d1)`

____ 5. A Button is positioned in a Frame. Its size is not affected when the Frame is resized. Which layout manager should be used?

- A. FlowLayout
- B. CardLayout
- C. North and South of BorderLayout
- D. East and West of BorderLayout

Part 4. READ PROGRAMS (25 points)

Write down the output of the following programs.

(1)

```
public class TestScope {
    private static int i = 1;
    private static int j = 2;
    public static void main(String[] args) {
        int i = 3;
        int k = 4;
        {
            int j = 5;
            System.out.println("i + j is " + i + j);
        }
        k = i + j;
        System.out.println("k is " + k);
        System.out.println("j is " + j);
    }
}
```

(2)

```
public class Test {
    public static void main(String[] args) {
        A a = new A(3);
    }
}

class A extends B {
    public A(int t) { System.out.println("A's constructor is invoked");}
}

class B {
```

```

    public B() { System.out.println("B's constructor is invoked"); }
}

```

(3)

```

class SuperReturnType{
    public String toString(){ return "superclass return type"; }
}
class SubReturnType extends SuperReturnType{
    public String toString(){ return "subclass return type"; }
}
class Superclass{
    SuperReturnType createReturnType() { return new SuperReturnType(); }
}
class Subclass extends Superclass{
    SubReturnType createReturnType()    { return new SubReturnType(); }
}
class CovarDemo{
    public static void main(String[] args)  {
        SuperReturnType suprt, subrt;
        suprt = new Superclass().createReturnType();
        subrt = new Subclass().createReturnType();
        System.out.println(suprt);
        System.out.println(subrt);
    }
}

```

(4)

```

class Person {
    private String name;
    Person(String aName) { name=aName;}
    public String getName() { return name; }
}
class Employee extends Person {
    private float basicSalary;
    private String employeeNumber;
    Employee(String aName, String aEmployeeNumber,float aBasicSalary) {
        super(aName);
        employeeNumber = aEmployeeNumber;
        basicSalary = aBasicSalary;
    }
    public String getEmployeeNumber() { return employeeNumber; }
    public float getBasicSalary() { return basicSalary; }
    public String toString(){
        return  getClass().getName() +" " +

```

```

        getName() + " " +
        " (employee number " + getEmployeeNumber() + ")\n" +
        "\thas a salary of " + getBasicSalary();
    }
    public static void main(String argv[]) {
        Secretary s = new Secretary("XU", "S008", 2500.0f);
        Manager m = new Manager("CHEN", "M007", 9000.0f, 2000.0f);
        System.out.println(s);
        System.out.println(m);
    }
}
class Manager extends Employee {
    private float allowance;
    Manager(String aName, String aEmployeeNumber,
        float aBasicSalary, float aAllowanceAmt) {
        super(aName, aEmployeeNumber, aBasicSalary);
        allowance = aAllowanceAmt;
    }
    public float getAllowance() { return allowance; }
    public String toString(){
        return super.toString() + "\n" +
        "\thas an allowance of " + getAllowance();
    }
}
class Secretary extends Employee {
    Secretary (String aName, String aEmployeeNumber, float aBasicSalary) {
        super(aName, aEmployeeNumber, aBasicSalary);
    }
}
}

```

(5)

```

public class TestException {
    public static void main(String[] args) {
        try {
            method();
            System.out.println("After the method call");
        }
        catch (RuntimeException ex) {
            System.out.println("RuntimeException in main()");
        }
        catch (Exception ex) {
            System.out.println("Exception in main()");
        }
    }
}

```

```

static void method() throws Exception {
    try {
        throw new RadiusException();
    }
    catch (RuntimeException ex) {
        System.out.println("RuntimeException in method()");
    }
    catch (Exception ex) {
        System.out.println("Exception in method()");
        throw ex;
    }
}
}

```

```

class RadiusException extends Exception{ }

```

Part 4.PROGRAMMING

1. Writing a Java GUI Application, to meet below requirements:

- JFrame name is “information”
- three JLabel components with text “Name”,”Number” ,”Class”
- three JTextField components
- One JButton, with name “确认”
- One JTextArea
- After input name,number,class information, then click the “确认”button, the TextArea will display the information you have inputed.

For example:

