第 6 周课堂作业

下面程序是否正确？如正确请写出运行结果，如错误请指出错误并修改程序代码。

1、程序 -1

```
public class Something {
        public static void main(String args[]){
                Something s = new Something();
                System.out.println("s.doSomething()returns " + doSomething());
        }

        public String doSomething() {
                return "Do something";
        }
}
```

s.doSomething()

答案：程序代码不正确。 main 方法是 static 静态方法，不能直接访问非静态方法 doSomething 。该程序可改成：
System.out.println("s.doSomething()returns " + s.doSomething());

2 、程序 -2

```
class Super {
  public int getLength() { return 4;}
}

public class Sub extends Super {
   public int getLength() { return 5;}

    public static void main(String args[] {
      Super sooper = new Super();
      Sub  sub = new Sub();
      System.out.println(
      sooper.getLength() + “,” + sub.getLength());
    }
        }
```

答：本题之类覆盖了父类的 getLength() 方法。所以父类对象和子类对象调用同名方法时，执行的是各自的方法，故本程序是正确的，执行结果是 4,5

3 、程序 -3

```
public class Something {

        private String s =

“”;

        int l = s.length();

}
```

有错。局部变量前不能放置任何访问修饰符，但 final 可以用来修改局部变量。因此应将 private 去掉。

4、程序 -4

```
1  class SuperClass
2  {
3     String nName = "SuperClass";
4     abstract void ShowTip();
5   }

6  class SubClassA extends SuperClass
7  {
8     String nName = "SubClassA";
9     String tip = "tip";
10     void ShowTip(String s)
11    {
12       System.out.println("This is " + nName);
13    }
14 }

15 public class SubClassB  extends  SubClassA
16 {
17    String c = "c";
18    public static void main(String[] args)
19    {
20       SuperClass a = new SuperClass();
21       System.out.println("This is " + a.nName);
22       a.ShowTip();
23       SubClassB b = new SubClassA();
24       b.ShowTip();
25       System.out.println("This is " + b.nName + "  " +
b.c);
26    }
27 }
```

5、程序 -5

```
public class Test
{
   private static int a = 1;
   private static int b = 2;

   public static int AddOne(int m)
   {
      b = a + m;
      return b;
   }

   public static int AddTwo(int a, int b)
   {
      b += AddOne(a);
      return b;
   }

   public static void main(String[] args)
   {
      b = AddTwo(3, 4);
      System.out.print(b);
   }
}
```

输出结果是 8

第 1 行有错，带有抽象方法的类应该是抽象类，修改为： abstract class SuperClass
第 10 行有错，该抽象方法实现是不带参数的，修改为： void ShowTip()
第 20 行有错，抽象类无法实例化，修改为： SuperClass a = new SubClassA();
第 23 行有错，类对象实例化类型不兼容，修改为： SubClassA b = new SubClassB();

6、程序 -6

```
public class Testson extends Test
{
   protected String sName = "Tom";
   public static void main(String args[])
   {
      Test t = new Testson ("John");
      System.out.println("Nice to meet you " + t.sName);
   }
   public Testson (String s)
   {
      super(s);
      System.out.println("How do you do?");
   }
   public Testson ()
   {
      this("I am Tom");
   }
}
class Test
{  protected String sName = "Jack";
   public Test ()
   {
      System.out.println("Hello!");
   }
   public Test (String s)
   {
      this();
      System.out.println("I am " + s);
   }
}
```

输出
Hello!
I am John
How do you do?
Nice to meet you Jack

7、程序 -7

```
class Creature {
   public void Eat(){
      System.out.println("Eat so good");
   }
}

class Animal extends Creature {
   public void Walk(){
      System.out.println("Animals Walk so good");
   }

   public void Eat(){
                  System.out.println("Animals Eat so good");
     }
}

public class Demo {
                  public static void main(String args[]){
     Creature one = new Animal();
     System.out.println(one.Eat() + "," + one.Walk());
                  }
}
```

程序代码不正确。变量 one 是 Creature 的对象，而 Creature 类中没有 Walk()

8、程序 -8

```
public class Test {
   public static void main(String[] args) {
     MyClass obj=new MyClass();
     obj.Info+=",World!";
     System.out.println(obj.Info);
   }
}

class MyClass {
    public String Info="Hello";
    public MyClass(String Info)
    {
        this.Info=Info;
    }
}
```

9、程序 -9

```
abstract class A {
    private abstract void f();
}

class B extendsA {
    int data;
    public B() {
    }

    public void f() {
        private int t = data;
        System.out.println(t);
    }
}
```

1. 第 2 行，abstract 的方法 f() 不能以 private 修饰，由于实现时为 public ，改为 public 。
2. 第 12 行，局部变量前不能放置任何访问修饰符，修改为 int t=data;

10 、程序 -10

```
class Parent {
   protected String sName = "Tom";
   protected String sCourse = "Java";

   public Parent() {
      System.out.println(sName + " like " + sCourse);
   }

   public Parent(String sn, String sc) {
      this();
      System.out.println(sn + " like " + sc);
   }
}

 public class Child extends Parent {
   protected String sName = "jack";
   protected String sCourse = "English";

   public Child(String sn, String sc) {
      super(sn, sc);
      System.out.println("What do you like?");
   }

   public Child() {
      System.out.println(sName + " like " + sCourse);
   }

   public static void main(String args[]) {
      Parent t = new Child("Max", "Math");
      System.out.println("I like " + t.sCourse);
   }
}
```

输出结果：

Tom like Java
Max like Math
What do you like?
I like Java

程序实现题

11、下面是平面图形类 Shape 的声明：

```
public abstract class Shape {

    public abstract double area();                    // 求图形面积

    public abstract double perimeter();  // 求图形周长

}
```

请实现其子类：**Rectangle**（矩形）、**Square**（正方形）、**Circle**（圆）。

提示：正方形应为矩形的子类