浙江大学 2006－2007 学年春季学期

《软件工程》课程期末考试试卷

开课学院：____计算机学院_____，考试形式：开卷，允许带_一本教科书__入场

考试时间：_2007_年_4_月_23_日， 所需时间：_120_分钟

考生姓名：_____________学号：___________________专业：____________教师：____________

题序	一	二	三	四	总 分
得分					
评卷人					

Answer Sheet

Part I				
1. ABC	**2. AD**	**3. ACD**	**4. BD**	**5. ABCD**
6. ACD	**7. BC**	**8. AB**	**9. BC**	**10. C**
Part II				
1. F	**2. T**	**3. F**	**4. T**	**5. T**
6. F	**7. F**	**8. F**	**9. F**	**10. T**
Part III				

1.

Size-oriented measures are computed by normalizing direct measures of the software engineering process (e.g. effort or defects) over the product size, measured in lines of code.

Function-oriented measures are indirect measures that are computed from measures of the information domain of a business application and an assessment of its complexity.

Size-oriented metrics are relatively easy to collect, but can present problems when component-based or visual programming methods are applied.

Function-oriented metrics can be determined much earlier in the software cycle, but are an abstraction that is open to interpretation.

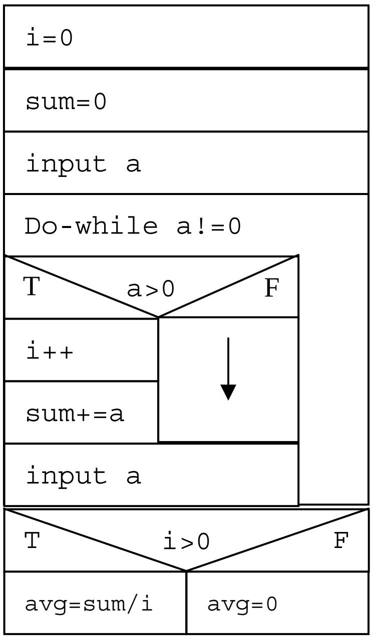
Developers need to measure so that they can tell whether they are improving or not. Without measurements this is extremely difficult to achieve.

2.

```
((A>B)&&(C==D)||(E<=G))

Define S = (A>B)&&(C==D).  Then the problem becomes (S || (E<=G)).
Lease cases needed: (T || T), (F || F).  That is, (T, <), (T, ==), (F, >).

For S being T: (T && T) = (>, ==).
For S being F: (T && F), (F && T). That is, (>, <), (>, >); (<, ==), (==, ==)

Hence the number is (S being T)*2 + (S being F) = 2 + 4 = 6.
```

3.

```
i=0;
sum=0;
input a;
while (a!=0) {
    if (a>0){
        i++;
        sum+=a;
    }
    input a;
}
if(i>0)
    avg=sum/i;
else
    avg=0;
```



4.

Classes provide an encapsulation (information hiding) mechanism by which data (attributes) have their access controlled by a set of operations. When properly implemented this yields systems with low coupling and high modularity.

Inheritance provides a mechanism by which changes to higher level classes can be propagated to lower level classes quickly.

Polymorphism reduces the effort required to extend an object system by enabling a number of different operations to share the same name.

Part IV

Any consistent answeres are acceptable.