M.M.Mahbubul Syeed

OHJ-3056 Software Engineering Methodology

Exam date: 17.5.2011, Tuesday

-Write your answer in English only, try to use a readable handwriting
-Answer on a separate paper, don't use this question paper
-No calculator
-There are five questions. Marks on the margin indicate full marks.

Marks

1. a. Briefly distinguish between the terms. 4
   i. Functional and Non functional requirements.
   ii. Software architecture and software design.
   iii. XP and Scrum.
   iv. *Interview* and *Business Architecture and key drivers* for capturing requirements.

   b. Calculate the MI (Maintainability Index) value for the following C program. 6

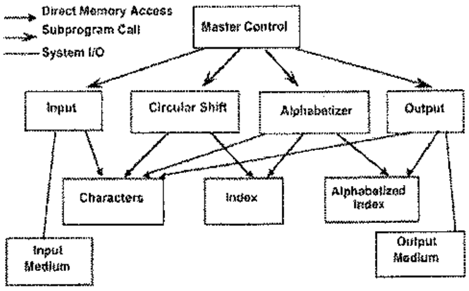
```
void main() {
if((*mode &0x30)== 0x30) scr=(char far*) 0xb0000000;
else { scr=(char far*) 0xb8000000;}
prevtimer=getvect(8);
setvect(8,mytimer);
keep(0,1000); }

void interrupt mytimer(){
int i, j, k;  char t[80];  ticks++;
if(ticks==18) {
ticks=0;
if(running==0) { running=1; }
for(i=0;i<25;i++){
for(k=0;k<=79;k++) {
t[k]=*(scr+i*160+k*2); }
k=0;
for(j=79;j>=0;j--) {
writechar(t[k],i,j,7);
k++;
} } }
running=0;
(*prevtimer)(); }

void writechar(char ch,int row,int col,int attr) {
*(scr+row*160+col*2)=ch;
*(scr+row*160+col*2+1)=attr; }
```

2. a. Please briefly describe which process model would be appropriate for the following scenario with proper justification of your answer. 4
   "Your project team is developing a new system which is fairly a large project. The team has adequate experience in the domain. But customers of this system are not experienced in using such system and require assistance to clarify their needs."

   b. Consider you are implementing a system according to following specification- "The KWIC index system accepts an ordered set of lines, each line is an ordered set of 6 words, and each word is an ordered set of characters. Any line may be *circularly shifted* by repeatedly removing the first word and appending it at the end of the line. The KWIC index system outputs a listing of all circular shifts of all lines in alphabetical order." A possible architecture for this requirement can be as follows,



Your task is to redesign this system using (i) layered architecture and (ii) repository architecture. You also need to make a comparison among these three (take into account the given architecture) architectural solutions in terms of performance, reuse and modifiability (functional, algorithm and data representation).

3. a. For COCOMO II model of effort estimation, discuss the followings, 5

   i. Three stages of a project development where COCOMO is used.
   ii. Basic model of COCOMO II.

   b. Determine whether or not each of the following statement is describing a risk. For each possible risk, describe (a) the type of risk and (b) possible risk management strategy. 5

   i. Stakeholders are not consistent about their requirements and changes occur frequently.
   ii. A new team is formed with newly recruited employees.
   iii. To aid the customer in identifying requirements, several prototypes are planned.
   iv. There was a gossip in the organization and the top management took quick and effective action against it.
   v. Estimated deadline cannot be met due to late delivery of required hardware.

4. Let's consider a hypothetical "Gas Station Control System" (or GSCS) that will be used to help manage an American-style gasoline or service station. Station provides following two services: 10

   - There is a small store that carries car parts. Inside the store is at least one cash register, operated by a cashier who is an employee of the gas station.
   - There are a number of gas pumps, at which customers can park their cars, interact with the system to pay via credit card, and then pump their own gas. Alternatively, the customer can pay for his or her gas via cash or credit card by going into the store and paying directly to the cashier.

Thus the GSCS has two main classes of users. The first is the cashier, who uses the GSCS to record purchases of car parts by customers. The GSCS must allow the cashier to enter the type and number of parts purchased, then computes the total purchase price and handles the payment. Customers purchasing gasoline are the second type of user. These customers interface with the system at the gas pump, by specifying the amount and type of gas they will buy, paying either at the pump or to the cashier, and then pumping the gas themselves.

The system also has to interact with other automated systems to perform its tasks. For example, in order to accept credit card payments, the GSCS must interface with a system maintained by the credit card company. The credit card system is responsible for checking that the customer's account is in good standing and can accommodate the amount of the purchase, and for debiting the customer's account and eventually reimbursing the gas station. The operation of these external systems is beyond the scope of the GSCS, although the GSCS needs to know how the external systems will communicate the success or failure of their tasks.

Your task is to come up with a use case diagram, one class diagram and a sequence diagram to design the GSCS system.

5. Consider the EMPLOYEE class bellow. The purpose of this class is to calculate employee benefits based on employee information (see the attribute list). Your task is to answer the following question- "Is there any design constrain violated in designing the class?" If your answer is yes, then what are those constrains violated and redesign the class to accommodate those. If your answer is no, then provide clear explanation to support your answer. 10

NOTE: you need not to be concerned about the attributes used in calculation. Only concentrate on the design.

```
Class EMPLOYEE{
        STRING* name;
        STRING* jobprofile;
        INTEGER* personalSkillLevel;
        INTEGER* birthDate;
        INTEGER* dayOfBirth;
        INTEGER* monthOfBirth;
        INTEGER* yearOfBirth;
        INTEGER* dayOfHire;
        INTEGER* monthOfHire;
        INTEGER* yearOfHire;
      public:
      void employeeBenifit();
      void calculatePersonalSkillLevel();
     Float computeSalary();  };
void employeeBenifit() {
   int age = (yearOfHire - yearOfBirth);
   float bonus = age * increment;
   float loanAmount = salaryDeposited + age * personalSkillLevel *2;
   float retirementFundAvailable = (age * .5) * salary deposited;      }
 void calculatePersonalSkillLevel() {   personalSkillLevel = age * jobdemand*.6; }
Float computeSalary() {
   calculatePersonalSkillLevel();
   return (Basic + personalSkillLevel*20);          }
```