

Shahjalal University of Science and Technology
Institute of Information and Communication Technology
Software Engineering
Final Examination, 3rd Year 2nd Semester
Course Code: SWE 331 Credits: 2 Course Title: Software Usability and Metrics
Total Marks: 30

Group A
[Answer all the questions]

1.

a) Suppose you provide a report containing the number of staff months necessary to build a product to your boss. But he can only afford a fraction of your proposed staff months. How will you handle this situation?

b) Consider the attribute, "number of bugs found," for software-testing processes. Define an absolute-scale measure for this attribute. Why is "number of bugs found" not an absolute scale measure of the attribute of program correctness?

2x2.5=5

2.

a) You have a small program with 10 simple inputs, 12 data file (trivial), and twelve outputs, each of average complexity. How many unadjusted function points would this be? Use the following table for reference:

2x5=10

TABLE 4.3 Function Point Complexity Ratings

Component	Simple	Average	Complex
Inputs (I)	3	4	6
Outputs (O)	4	5	7
Data Files (F)	7	10	15
Interfaces (N)	5	7	10
Inquiries (Q)	3	4	6

- b) What do you understand by a Decision Maker Model? Explain with relevant diagram.

Group B
[Answer all the questions]

3.

a) Suppose you have a question “How user-friendly is your system to the user”? Now design two metrics to answer this question.

b) Suppose that we could classify every software failure as either a) syntactic, b) semantic, or c) system crash. Suppose additionally that we agree that every system crash failure is more critical than every semantic failure, which in turn is more critical than every syntactic failure. Use this information to define two different measures of the attribute of criticality of software failures. How are these measures related? What is the scale of each?

2x2.5=10

4.

2x5=10

- a) Find the LOC, CC and Weighted Information flow for the following code.

```
void merge(int arr[], int l, int m, int r) {
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;
    int L[n1], R[n2];
    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];
    i = 0;
    j = 0;
    k = l;
    while (i < n1 && j < n2)
    {
        if (L[i] <= R[j])
        {
            arr[k] = L[i];
            i++;
        }
        else
        {
            arr[k] = R[j];
            j++;
        }
        k++;
    }
    while (i < n1)
    {
        arr[k] = L[i];
        i++;
        k++;
    }
    while (j < n2)
    {
        arr[k] = R[j];
        j++;
        k++;
    }
    return;
}
```

References:

1. CC = Binary Decisions + 1
2. Weighted IFC = length * (fanin * fanout)²

- b) Suppose that "complexity" of individual software modules is ranked (according to some specific criteria) as one of the following:
{ trivial, simple, moderate, complex, very complex, incomprehensible }

Let M be any measure (in the representation sense) for this notion of complexity, and let S be a set of modules for each of which M has been computed.

State the following questions.

- i. You want to indicate the average complexity of the modules in S. How would you do this in a meaningful way? (Briefly explain your choice.)
- ii. Explain why it is not meaningful to compute the mean of the Ms. (You should construct a statement involving means that you can prove is not meaningful.)
- iii. Give two examples of criteria that might be used to enable an assessor objectively to determine which of the complexity values a given module should be.