Software Engineering
Preliminary Examination
February 2009

Below is a list of sample exam questions to help you prepare for the Software Engineering Preliminary Exam. The questions provided do not cover every topic addressed by the exam. Be sure to review all topics identified in the preliminary exam syllabus prior to attempting the software engineering preliminary exam.

(1) Provide a complete explanation of the differences between the terms **class**, **object**, and **client**.

(2) Given the code segment below, answer the following **three questions**.

```
class  Alpha                    class  Bravo  :  public  Alpha
{                               {
  public:                         public:
    Alpha();                       Bravo();
    void Charlie();                void Foxtrot();
  private:                        private:
    int  Delta;                    float Gamma;
    int  Echo();                   void Helo()
};                              };
```

If a variable of type **Bravo** is created, which, if any, constructors are called and in what order?

An object of type **Alpha** has _____ member functions directly accessible by a client program?

An object of type **Bravo** has _____ data attributes directly accessible by a client program?

(3) In the context of C++, explain the differences between direct and indirect addressing.
You must include C++ examples of each, in addition to your explanation, to receive full credit.

(4) Describe the ***Slicing Problem*** and how it may be overcome in C++.
Be sure to include sample C++ code that illustrates both the problem and the solution to the problem.

(5) For the code segment below, describe the role of the **&** operator on **Line2** and the ***** operator on **Line3**.

```
int  x = 5;          // Line 1
int*  y = &x;        // Line 2
cout << *y;          // Line 3
```

(6) Describe the difference between static and dynamic binding.

(7) What term is used to describe to pool of memory locations used for dynamic allocation/deallocation?

(8) Give two reasons why one might wish to pass an object by reference.

(9) Suppose that class ***A*** has a private member ***X***, a protected member ***Y***, and a public member ***Z***.
A new class ***B*** is derived from ***A***. Circle the visibility/accessibility of each inherited member within ***B***.

class B : public A	**member X**:	public	protected	private	inaccessible
{					
…	**member Y**:	public	protected	private	inaccessible
};					
	member Z:	public	protected	private	inaccessible

(10) Given the following code segment, answer the two questions below:

```
int*  ptr1 = new int;          // Line 1
int*  ptr2 = new int;          // Line 2
*ptr2 = 44;                    // Line 3
*ptr1 = *ptr2;                 // Line 4
ptr1 = ptr2;                   // Line 5
delete ptr2;                   // Line 6
ptr2 = NULL;                   // Line 7
```

Which line creates an **inaccessible object**?

Which line creates a **dangling pointer**?

You must correctly explain your answers to receive full credit.

(11) Overload the **<<** operator for the **Fraction** class. Example: the fraction three-fourths should output as **3 / 4**

```
class   Fraction
{
  public:
     Fraction(int num, int denom) { n = num; d = denom; }

  private:
     int  n;      // Numerator
     int  d;      // Denominator
};
…
Fraction  X(3, 4);
Fraction  Y(5, 6);
cout  <<  X  << " + "<< Y ;   // Sample use of overloaded << operator
```

(12) Explain the differences between **reference types** and **pointer types** in C++.

(13) Write a C++ template function that swaps the values stored within two variables of some arbitrary data type. Be sure to use correct syntax and semantics. What underlying assumption are you making as part of your implementation?

(14) What is a ***friend function*** and why would one choose to create a friend function as opposed to a class member function?

(15) Give two example of ***Generic Programming*** in C++.

(16) Compare the efficiency of iterative and recursive implementations of the factorial calculation (**n!**) in terms of both execution speed and memory usage.

(17) What is a ***Virtual Function*** and how is it used in C++?

(18) This problem uses the declarations of classes **A** and **B** and functions below:

```
class A
{
  public:
    A();
    virtual void Mystery();
    void Paradox();
};

A::A()
{
   // Empty Constructor
}

void A::Mystery()
{
  cout << "MysteryA";
}

void A::Paradox()
{
  cout << "ParadoxA";
}

void Enigma(A param)
{
  param.Mystery();
  param.Paradox();
}
```

```
class B : public A
{
   public:
     B();
     void Mystery();
     void Paradox();
};

B::B()
{
  // Empty Constructor
}

void B::Mystery()
{
   cout << "MysteryB";
}

void B::Paradox()
{
   cout << "ParadoxB";
}

void Unknown(A& param)
{
  param.Mystery();
  param.Paradox();
}
```

Given object **objA** of type **A** and **objB** of type **B**, what output is written to **stdout** for the following function calls?

```
objA.Mystery();
```
_____

```
objB.Mystery();
```
_____

```
Enigma(objA);
```
_____

```
Enigma(objB);
```
_____

```
Unknown(objA);
```
_____

```
Unknown(objB);
```
_____

(19) Explain the differences between ***Big-O***, ***Big-Omega***, ***Big-Theta***, *and* ***Little-O*** notations as they relate to analysis of algorithms.

(20) What is meant by the term ***NP complete***?

(21) List two problems that are ***NP complete*** and provide a brief description of each problem.

(22) Explain the differences between a **Depth-First Search** and a **Breadth-First Search**.

(23) Compare the efficiency of **Bubble Sort** and **Selection Sort** in terms of the number of comparisons made and the number of swaps made in the Worst Case assuming **N** elements are being sorted.

(24) Perform a ***Big-O*** complexity analysis of the following code segment. Assume all variables are of type **int**.

```
int M = 0;
int N;
/* N is set here */
for(int k = 0; k < log(N); k++)
{
  for(int j = N; j >= 0; j--)
    M = M *j – k;
}
```

(25) In the ***Best Case***, what is the complexity of inserting a new element into a **Binary Search Tree** (**BST**) that has N elements. What is the ***Worst Case*** complexity of the insert operation change in a **BST**?

(26) Insert the following list of integers in order {15, 3, 22, 97, 18, 7, 55, 30, 19, 10, 12, 77} into a **Maximum Heap** and draw the resulting structure. Delete the heap's root node and redraw the resulting **Maximum Heap**.

(27) Describe the difference between the "is-a", "has-a", and "uses" relationships in object-oriented development, and draw a sample UML diagram that illustrates "is-a", "has-a", and "uses" relationships between classes.

(28) Describe the differences between Use Case Diagrams, Class Diagrams and Sequence Diagrams in UML. Be sure to draw examples of each diagram as par of your solution.

(29) What is the cyclomatic complexity of the following code:

```
for(i=0;0<10;i++)
   for(j=0;j<i;j++)
   {
      n = n*j + i;
       switch(n)
       {
          case 0:  n = 2;      break;
          case 1:
          case 2:  n = n % 8;  break;
```

```
            case 3:
            case 5:  n = n + 3;  break;
            default:             break;
        }
    }
```

(30) Write a state diagram for the above piece of code.

(31) Describe the differences between Phases and Disciplines (Workflows) in the Unified Process.

(32) Describe three different models of development and the kinds of applications where they are useful

(33) What three problems lead to the development of the Software Engineering field?

(34) Describe the five levels of the Capability Maturity Model including the KPAs addressed at each level. What is the minimum CMM level required for Department of Defense work?

(35) For each risk identified and described during a Risk Analysis, three important factors must be determined. Describe each of these three factors.

(36) Describe the roles identified in the Chief Programmer team organization strategy and draw an organization chart that illustrates the managerial hierarchy.

(37) Describe a Fagan-style inspection as it relates to software development. Include a description of the inspection roles and responsibilities. How does an inspection differ from a walkthrough?

(38) Explain the differences between the terms Coupling and Cohesion with respect to software engineering. What forms of Coupling and Cohesion are desirable and why?

(39) What is ***Brook's Law***?

(40) You are going to have to do a time, sloc, and cost estimate for a project that has not started. Describe three different estimating methods and where they are applicable (what are the pre-conditions, what can they tell you, what is their precision).

(41) Describe Earned-Value Planning as it relates to software development.

(42) List two different Requirements Elicitation techniques.

(43) Describe the differences between Black Box and Clear Box testing.

(44) Describe the three basic integration strategies and the roles of drivers and stubs in this context.

(45) What is the difference between Statement Coverage, Branch Coverage, and Path Coverage?

(46) You have been hired as a test engineer for a large mission critical system. Describe three different test methodologies and how they might be useful.

(47) Describe a procedure and tool for testing user interfaces.

(48) What configuration management problems are addressed by UML Deployment Diagrams.

(49) What configuration management problems are addressed by a typical source control system.

(50) Compare and constrast the Waterfall Process and Spiral Development. Be sure to include the advantages and disadvantages of each process.

(51) Explain the differences between the Stack, Queue, and List abstract data types.

(52) Compare the efficiencies of the Stack operations Push, Pop, and Top using the Big-O notation for both the array implementation and the linked list implementation.

(53) Explain the differences between an array and a vector in C++.

(54) What error detection technique is most efficient in terms of the number of defects identified per hour with respect to non-compile errors? Justify your answer.

(55) Give three examples of defect prevention techniques. Justify your answers.

(56) What is the difference between unit test, integration test, system test, and acceptance test?

(57) Describe two methods for representing an arbitrary weighted graph in C++.

(58) Describe the exception handling mechanism built into C++. Give an example of its use in trapping a divide-by-zero error.

(59) List three software engineering metrics and describe their use in managing the quality of a software product.

(60) What is a hash function and what problem was the hashing technique developed to solve? You must include a sample hash function as part of a code example to receive full credit.

April 2009

Software Engineering
Preliminary Exam

(1) [6 points] What ***three major problems*** that led to the development of the software engineering field?

(2) [8 points] What is the purpose of a ***rapid prototype***? What are the key issues related to **reuse** of a rapid prototype?

(3) [6 points] In the **average case**, what is the ***Big-O*** complexity of inserting a new element into a **binary search tree** that has **N** elements? What is the ***Big-O*** complexity of inserting an element into a binary search tree in the **best case**? ***You must explain how you arrived at your answers for both questions.***

(4) [10 points] What is ***polymorphism***? Provide a source code example illustrating polymorphism in C++.

(5) [10 points] Describe the ***slicing problem*** and provide sample source code that illustrates how the slicing problem is overcome in C++.

(6) [10 points] Draw the ***binary search tree*** containing the following values
{15, 7, 9, 21, 44, 30, 33, 29, 10, 1, 17}
inserted in the order shown (from left to right).

(7) [10 points] Describe ***Equivalence Class Partitioning*** and ***Boundary Value Analysis***? Which technique typically generates a larger set of tests?

(8) [10 points] Describe the role of the ***Sequence Diagram*** in UML, and draw an example of a Sequence Diagram.

(9) [10 points] Describe the **five major steps** of a ***Fagan-style inspection***. Name two key differences between an inspection and a walkthough which make an inspection more rigorous.

(10) [6 points] Suppose you are a customer purchasing a small but safety-critical software package. The vendor gives you choice with respect to the test coverage criteria that will be used: statement coverage, branch coverage, or path coverage. Given that the vendor will

guarantee the level of test coverage, which level would you select? Explain your selection.

(11) [4 points] What are two major weakness of the ***Waterfall process***?

(12) [6 points] Draw the graph described by the following adjacency matrix

	A	**B**	**C**	**D**
From A	1	0	0	0
B	1	0	0	0
C	1	1	0	1
D	1	0	0	0

(13) [4 points] What is the ***cyclomatic complexity*** of the above graph?

Software Engineering
October 2009

Problem 1. [5 points]

In the 80's and 90's dozens of new programming language were developed. In the past few years, hardly any new languages have been developed. What do you think the reason for this is?

Problem 2. [5 points]

The newest computers are coming out with 6 core, 4 chip motherboard (24 cores total) architectures. Describe the difference between threads and processes. Describe 3 issues associated with distributing them amongst a single core, multiple cores on a single chip and multiple cores on multiple chips.

Problem 3 [10 points]

Write the output produced by the following C++ program on the lines provided.

```
#include <iostream>
using namespace std;

class AAA
{
 private:
  int A;
 public:
  AAA() { A = 3; }
  AAA(int a) { SetA(a); }
  void SetA(int a) { A = a; }
  int GetA() const { return A; }
  virtual void Print() { cout << GetA() << endl; }
};

class BBB : public AAA
{
 private:
  int B;
 public:
  BBB() : AAA(8) { B = 6; }
  BBB(int b)  { SetB(b); }
  void SetB(int b) { B = b; }
  int GetB() const { return B; }
  void Print() { cout << GetA() << "   " << GetB() << endl; }
};

void Write(AAA param1)
{
  param1.Print();
}

void Display(AAA& param1)
{
  param1.Print();
}

int main()
{
  AAA obj1;
  AAA obj2(4);
  BBB obj3;
  BBB obj4(7);
  obj1.SetA(2);
  obj2.SetA(obj3.GetB());

  Write(obj2);                  // Output # 1 = _____

  Write(obj4);                  // Output # 2 = _____
  obj1 = obj3;
  obj1.Print();                 // Output # 3 = _____
  obj3 = obj4;
  obj3.Print();                 // Output # 4 = _____

  Display(obj2);                // Output # 5 = _____

  Display(obj4);                // Output # 6 = _____
  return 0;
} // End main()
```

Problem 4. [8 points]

For the program listed in ***Problem 3***, draw a ***UML class diagram*** in the space below.

Problem 5. [2 points]

Which UML diagram documents the message passing required to realize a Use Case?

Problem 6. [10 points]

For the program below, write the outputs on the corresponding lines provided.

```
// Assume all values including memory addresses are output in standard decimal
// notation.  Write the output of the following program on the lines provided.
#include <iostream>
using namespace std;

void DoSomething(int a, int* b);

int main()
{
  int x = 4;        // Assume the variable x is stored at address 1000
  int y = 5;        // Assume the variable y is stored at address 1200
  int* p1 = &x;     // Assume the variable p1 is stored at address 1400
  int* p2, p3;      // Assume the variable p2 is stored at address 1600
                    //    and the variable p3 is stored at address 1800


  cout << x << endl;           // Output # 1 = _____________________________


  cout << y << endl;           // Output # 2 = _____________________________


  cout << p1 << endl;          // Output # 3 = _____________________________

  p2 = &y;
  cout << p2 << endl;          // Output # 4 = _____________________________

  *p1 = *p1 + y;
  cout << p1 << endl;          // Output # 5 = _____________________________


  cout << *p1 << endl;         // Output # 6 = _____________________________

  p1 = p2;
  cout << p1 << endl;          // Output # 7 = _____________________________


  cout << p2 << endl;          // Output # 8 = _____________________________

  p3 = *p2;
  p3++;
  cout << p3 << endl;          // Output # 9 = _____________________________


  DoSomething(x, &y);
  cout << y << endl;           // Output #10 = _____________________________

  return 0;
}  // End main()

void DoSomething(int a, int* b)
{
  *b = a;
}  // End DoSomething()
```

Problem 7. [5 points]

Agile development processes are popular amongst developers since they are not heavyweight, document-focused processes. List **five** of the twelve underlying principles of Agile software development as described in the **Agile Manifesto.**

Problem 8. [5 points]

How does the organization of modern programming teams with separate ***Project Manager*** and ***Team Leader*** roles address the shortcomings of the classic ***Chief Programmer*** team organizational strategy?

Problem 9. [10 points]

Define the term ***task hour*** as used in the context of ***Earned-Value Planning.***

Explain the difference between **Planned Value** and **Earned Value.**

If a development team wishes to see ***continuous progress each week*** when using the **Earned-Value Method** of project tracking, what criteria should the team use when defining the granularity of tasks included in their earned value plan?

Problem 10. [5 points]

What is the purpose of a ***Traceability Analysis*** in the software development process?

Problem 11. [5 points]

In the context of Fagan-style Inspections, what is meant by the term Inspection Yield? How is Inspection Yield data used in a CMM Level 5 software process?

Problem 12. [5 points]

What is meant by the phrase **"Quality is Free"** in the context of software development?

Problem 13. [10 points]

You have been asked to use the construct both ***top-down*** and ***bottom-up*** estimates of the effort required to develop a new software product. Which techniques will you use and why?

Problem 14. [5 points]

What is the difference between **verification** and **validation**?

Problem 15. [10 points]

What are the advantages and disadvantages of object-oriented software development with respect to the complete software life cycle? Be sure to address **inheritance, polymorphism** and **dynamic binding** in your answer.

Software Engineering Preliminary Exam

2010 Spring

(1) [8 points] There are ***twelve core practices*** within **Extreme Programming (XP).** Describe four of the twelve core practices and how each of these practices contributes to the classification of **XP** as an "**Agile**" process.

(2) [5 points] In Kent Beck's original formulation of **Extreme Programming (XP)**, what variable is used in an attempt to control the variables **cost**, **schedule, and quality** and **why**?

(3) [5 points] You have been asked to lead a five-person software development team that will implement a small but safety critical software package. Which software process would you select and why?

(4) [3 points] What is the **Moving-Target Problem** and how does it resemble and/or differ from **Feature Creep**?

(5) [4 points] What are **two advantages** and **two disadvantages** of the object-oriented approach to software construction?

(6) [4 points] What is ***McCabe's Cyclomatic Complexity*** and what are the implications of a large cyclomatic complexity on the software testing process?

(7) [8 points] What is **hashing**? Discuss the best case and worst case performance of hashing. Provide an example of a hash function.

(8) [5 points] Suppose that a **maximum heap** is implemented using a one-dimensional array of integers. Show the contents of the heap array **after** the following six integers have been inserted into the heap in order from left to right { 5, 4, 10, 15, 21, 7}. Show the final contents of the heap array **after** the value 21 has been removed from the heap.

(9) [6 points] What are the differences between **public**, **protected**, and **private** members in C++? Provide a C++ example to illustrate the differences.

(10) [8 points] You are required to develop a risk analysis for a software system for a real time controller for a nuclear reactor.

A) List five risks that could significantly affect the project and mitigation techniques.

B) List three methods for estimating the cost of the project. Provide details

(11) [15 points] Choose one of the following:

A) Write a recursive sort algorithm in the language of your choice.

B) Write a recursive search algorithm in the language of your choice.

(12) [5 points] Linux systems appear to be much more secure than Windows systems. Provide five reasons why this might be true.

(13) [6 points] List three sorting algorithms. What is the computational complexity of each algorithm?

(14) [10 points] Describe the work products of a **waterfall** model product. Explain where in the design cycle each is produced.

(15) [8 points] Describe the difference between functional and non-functional requirements. Give two examples of each.

Software Engineering Preliminary Exam

2010 Spring

(1) [8 points] There are ***twelve core practices*** within **Extreme Programming (XP).** Describe four of the twelve core practices and how each of these practices contributes to the classification of **XP** as an "**Agile**" process.

(2) [5 points] In Kent Beck's original formulation of **Extreme Programming (XP)**, what variable is used in an attempt to control the variables **cost**, **schedule, and quality** and **why**?

(3) [5 points] You have been asked to lead a five-person software development team that will implement a small but safety critical software package. Which software process would you select and why?

(4) [3 points] What is the **Moving-Target Problem** and how does it resemble and/or differ from **Feature Creep**?

(5) [4 points] What are **two advantages** and **two disadvantages** of the object-oriented approach to software construction?

(6) [4 points] What is ***McCabe's Cyclomatic Complexity*** and what are the implications of a large cyclomatic complexity on the software testing process?

(7) [8 points] What is **hashing**? Discuss the best case and worst case performance of hashing. Provide an example of a hash function.

(8) [5 points] Suppose that a **maximum heap** is implemented using a one-dimensional array of integers. Show the contents of the heap array **after** the following six integers have been inserted into the heap in order from left to right { 5, 4, 10, 15, 21, 7}. Show the final contents of the heap array **after** the value 21 has been removed from the heap.

(9) [6 points] What are the differences between **public**, **protected**, and **private** members in C++? Provide a C++ example to illustrate the differences.

(10) [8 points] You are required to develop a risk analysis for a software system for a real time controller for a nuclear reactor.

A) List five risks that could significantly affect the project and mitigation techniques.

B) List three methods for estimating the cost of the project. Provide details

(11) [15 points] Choose one of the following:

A) Write a recursive sort algorithm in the language of your choice.

B) Write a recursive search algorithm in the language of your choice.

(12) [5 points] Linux systems appear to be much more secure than Windows systems. Provide five reasons why this might be true.

(13) [6 points] List three sorting algorithms. What is the computational complexity of each algorithm?

(14) [10 points] Describe the work products of a **waterfall** model product. Explain where in the design cycle each is produced.

(15) [8 points] Describe the difference between functional and non-functional requirements. Give two examples of each.