

This print-out should have 12 questions, check that it is complete. Multiple-choice questions may continue on the next column or page: find all choices before making your selection. The due time is Central time.

EE319K Chapter 7 homework

001 (part 1 of 1) 10 points

What property of a debugging activity best describes the measure to which the debugging code itself affects the software it is trying to evaluate?

1. Stabilization.
2. None of these choices is correct.
3. Modularity.
4. Static efficiency.
5. Nonintrusiveness.
6. Hierarchical.

002 (part 1 of 1) 10 points

Which is the general term for software added to a system for the purpose of debugging?

1. Monitor.
2. Breakpoint.
3. Scanpoint.
4. None of these choices is correct.
5. Subroutine.
6. Instrument.

003 (part 1 of 1) 10 points

Consider the situation where you are trying to measure the number of characters processed per second. Assume the correct rate is 1000 characters per second, but it takes so much time to run the debugging code that the processing rate actually drops to 500 characters

per second. What term describes this debugging mistake?

1. Unstable.
2. Overflow.
3. Dropout.
4. Intrusive.
5. Underflow.
6. None of these choices is correct.

004 (part 1 of 1) 10 points

What is a **logic analyzer**?

1. Software application to analyze digital logic functions.
2. None of these choices is correct.
3. A digital voltmeter.
4. A device that emulates the microcomputer.
5. An analog oscilloscope.
6. A multiple channel digital storage scope with many ways to trigger.

005 (part 1 of 1) 10 points

What is an **emulator**?

1. A debugging technique used to implement breakpoints.
2. Software application that simulates the hardware and software functions of the microcomputer.
3. A software application used to provide debugging functions.
4. A hardware debugging tool that recreates the input/output signals of the processor

chip.

5. None of these choices is correct.
6. A multiple channel digital storage scope with many ways to trigger.

006 (part 1 of 1) 10 points

What is the **background debug module (BDM)**?

1. A **BDM** is a hardware built into the processor itself and an external POD. Together they provide debugging features such as observing memory while the program is running.
2. None of these choices is correct.
3. A **BDM** is a multiple channel digital storage scope with many ways to trigger.
4. A **BDM** is a software application, which runs in the background, that simulates the hardware and software functions of the microcomputer.
5. A **BDM** is software code added to the program to visualize time-dependent execution, which runs in the background.
6. A **BDM** is a hardware debugging tool that recreates the input/output signals of the processor chip.

007 (part 1 of 1) 10 points

What format field should you use to display the memory contents of three consecutive signed 16-bit integers in decimal format?

1. 3+H
2. +d,+d,+d
3. None of these choices is correct.
4. 3+D

5. 3+d

6. D,D,D

008 (part 1 of 1) 10 points

To verify the proper functionality of a subroutine, we need to record its input and output parameters. In particular we write a special main program that provides a known and repeatable sequence of inputs to the subroutine under test. In this way, each time the test subroutine is changed, we can be sure the change in output values is caused by the software modification and not due to a change in input values. What is this debugging procedure called?

1. Monitor.
2. Performance debugging.
3. Nonintrusive debugging.
4. None of these choices is correct.
5. Stabilization.
6. Emulation.

009 (part 1 of 1) 10 points

What is the major difference between **performance debugging** and **functional debugging**?

1. None of these choices is correct.
2. Performance debugging studies the time behavior, and functional debugging tests if the proper output values are produced.
3. Performance debugging evaluates the interactive between functions, and functional debugging measures the accuracy of the system.
4. Functional debugging studies the time behavior, and performance debugging tests if the proper output values are produced.

5. Functional debugging evaluates the interactive between functions, and performance debugging measures the accuracy of the system.

6. They are two names that mean the same thing.

010 (part 1 of 1) 10 points

What is the difference between **scanpoint** and **breakpoint**?

1. A breakpoint executes a software interrupt **swi**, and scanpoint executes a software trap **trap**.

2. They are two names that mean the same thing.

3. None of these choices is correct.

4. A scanpoint studies the time behavior, and a breakpoint tests if the proper output values are produced.

5. A breakpoint causes the program to stop, and scanpoint records strategic information without stopping execution.

6. A scanpoint causes the program to stop, and breakpoint records strategic information without stopping execution.

011 (part 1 of 1) 10 points

Which method can be used to measure the **execution speed** of a subroutine?

1. First determine how many times each assembly language instruction in the subroutine is typically executed. E.g., instructions that are not usually executed get a count of zero. Then, multiply these counts by the corresponding number of cycles each instruction takes to execute. Add them up to get the typical cycle count for each run through the subroutine. Lastly, multiply by the time per cycle for this microcomputer.

2. Run this program, and record the value in Reg D. Assume **before** is a global variable.

```

ldd  TCNT
std  before
bsr  funct
ldd  TCNT
subd before

```

3. Attach an unused output pin, PA0, to an oscilloscope. Run this main program, and measure the pulse width of PA0 using the scope.

```

main  lds    #$0C00
      bset  DDRA, #$01
loop  bset  PORTA, #$01
      bsr   funct
      bclr  PORTA, #$01
      bra   loop

```

4. All of these choices is correct.

012 (part 1 of 1) 10 points

Which answer best describes **profiling**?

1. None of these choices is correct.

2. **Profiling** is the analysis of the call graph creating a profile of which functions call which other functions.

3. **Profiling** involves the dynamic measurements of how fast a subroutine executes.

4. **Profiling** is the creation of a profile of the computing resources required of the software.

5. **Profiling** is the analysis of the data flow graph creating a profile of the data values as they are processed.

6. **Profiling** is the collection of information that involves which functions were executed, when they were executed, and what data were they processing.