Pre-Lecture 10

Due Sep 23 at 9am **Points** 14 **Questions** 12

Available until Sep 23 at 9am Time Limit None Allowed Attempts 2

Instructions

Take this quiz after you have watched the required videos and/or read the associated sections of the textbook. See <u>Lecture 10: Memory hierarchy</u>.

You may attempt this quiz twice. Incorrect responses are marked after each attempt. Correct answers are revealed at the start of class for this lecture.

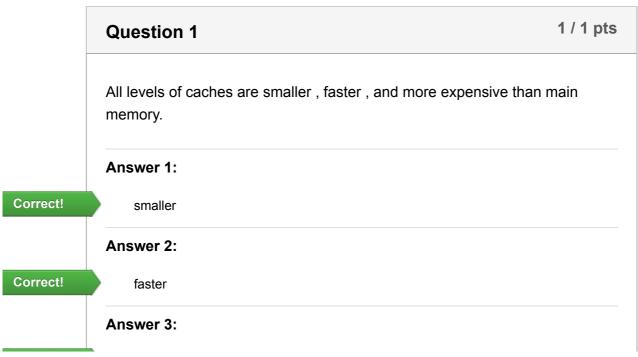
Carefully note the deadline for responses. Submissions are not accepted after the deadline, and there is no grace period.

This quiz was locked Sep 23 at 9am.

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	9 minutes	14 out of 14
LATEST	Attempt 2	9 minutes	14 out of 14
	Attempt 1	32 minutes	11.25 out of 14

Score for this attempt: **14** out of 14 Submitted Sep 22 at 11:53pm This attempt took 9 minutes.



Correct!

more expensive

	Question 2	1 / 1 pts
	A cache block that holds multiple data items and/or instructions supp what kind of locality?	orts
	cache	
Correct!	spatial	
	temporal	

Question 3 1 / 1 pts

Consider the following C code fragment:

```
for(i = 0; i < M; i++)
for(j = 0; j < N; j++)
  result += a[i] * b[i][j];</pre>
```

In the inner loop, array a exhibits temporal locality while array b exhibits spatial locality.

Answer 1:

Correct!

temporal

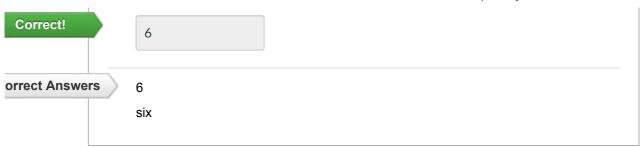
Answer 2:

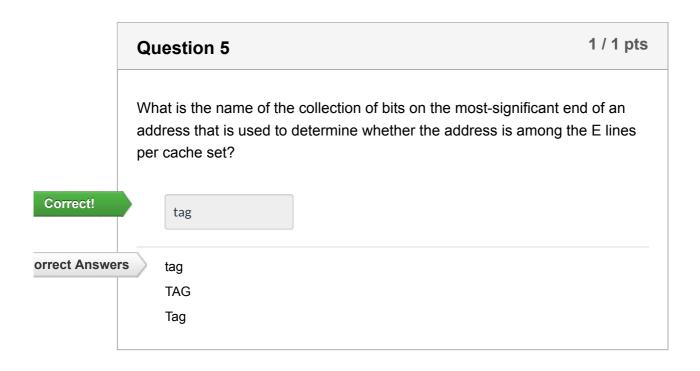
Correct!

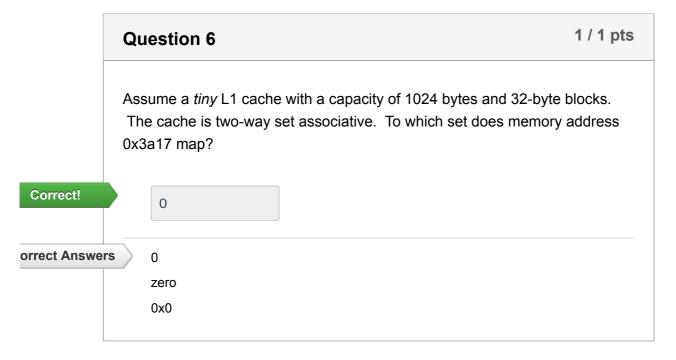
spatial

Question 4 1 / 1 pts

Consider an L1 cache with 64-byte blocks. How many bits on the least-significant end of an address are used to determine the block offset?





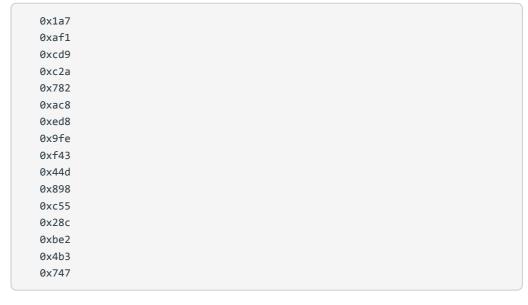


Question 7 1 / 1 pts

Recall that a fully-associative cache is one in which all lines reside in the same, single set.

Assume a *minuscule* fully-associative L1 cache with a capacity of 256 bytes and 16-byte blocks. Also assume that the cache currently contains addresses with the following tags and all valid bits set to 1.

Tags of addresses currently residing in cache:

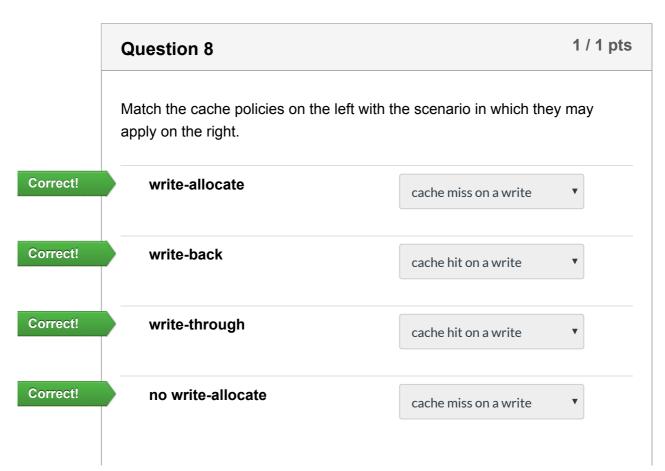


True or false: A memory access with address 0x1234 hits in this cache.

True

Correct!

False



Other Incorrect Match Options:

- · cache miss on a read
- · cache hit on a read

Question 9 1 / 1 pts

Suppose that we have an L1 cache of this configuration:

- B = 32 bytes
- S = 64
- E = 1
- C = 2048 bytes

What is the cache miss rate (as a percentage) when we execute the following C code? Assume that the grid data structure is aligned on a cache block boundary in memory and that the cache is cold.

```
struct {
   double x;
   double y;
} grid[16][16];

for(i = 0; i < 16; i++)
   for(j = 0; j < 16; j++)
     total_x += grid[i][j].x;</pre>
```

Correct!

50

orrect Answers

50

50%

fifty

fifty percent

50 %

Question 10 1 / 1 pts

Suppose that we have an L1 cache of this configuration:

- B = 32 bytes
- S = 64
- E = 1

• C = 2048 bytes

What is the cache miss rate (as a percentage) when we execute the following C code? Assume that the grid data structure is aligned on a cache block boundary in memory and that the cache is cold.

```
struct {
  double x;
  double y;
} grid[16][16];

for(i = 0; i < 16; i++)
  for(j = 0; j < 16; j++)
    total_x += grid[j][i].x;</pre>
```

Correct!

100

orrect Answers

100

100%

100 %

one hundred

one hundred percent

one-hundred

one-hundred percent

Question 11 1 / 1 pts

Suppose that we have an L1 cache of this configuration:

- B = 32 bytes
- S = 64
- E = 1
- C = 2048 bytes

What is the cache miss rate (as a percentage) when we execute the following C code? Assume that the grid data structure is aligned on a cache block boundary in memory and that the cache is cold.

```
struct {
  double x;
  double y;
} grid[16][16];

for(i = 0; i < 16; i++)
  for(j = 0; j < 16; j++) {
    total_x += grid[i][j].x;</pre>
```

Question 12 3 / 3 pts

Consider the following C code fragment that does matrix transpose:

```
int i, j;
for(i = 0; i < M; i++)
  for(j = 0; j < N; j++)
   B[j][i] = A[i][j];</pre>
```

Complete the following optimized version of matrix transpose. Assume that values for V and W are chosen such that a WxV block of matrix B and a VxW block of matrix A fit in cache at the same time.

```
int i, j, k, l;
for(i = 0; i < M; i+=V)
  for(j = 0; j < N; j+=___)
  for(k = ___; k < ___; k++)
    for(l = ___; l < j+W; l++)
    B[___][__] = A[___][__];</pre>
```

Blank 1: W

Blank 2: i

Blank 3: i+V

Blank 4: j

Blank 5:

Blank 6: k

	Blank 7: k
	Blank 8:
	(NOTES: Blanks are ordered top to bottom, left to right. <i>Exact</i> C code is expected in the blanks. Avoid including any unnecessary blank spaces in your answers.)
	Answer 1:
Correct!	W
	Answer 2:
Correct!	i
	Answer 3:
Correct!	i+V
orrect Answer	i + V
	Answer 4:
Correct!	j
	Answer 5:
Correct!	
	Answer 6:
Correct!	k
	Answer 7:
Correct!	k
	Answer 8:
Correct!) I

Quiz Score: 14 out of 14