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 **Lecture 10: Memory hierarchy**.

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.

Question 1 — 1 / 1 pts

All levels of caches are smaller , faster , and more expensive than main memory.

Answer 1:

Correct! smaller

Answer 2:

Correct! faster

Answer 3:

Correct! more expensive

Question 2

1 / 1 pts

A cache block that holds multiple data items and/or instructions supports what kind of locality?

- 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];
```

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?

Correct!

6

orrect Answers

6

six

Question 5

1 / 1 pts

What is the name of the collection of bits on the most-significant end of an address that is used to determine whether the address is among the E lines per cache set?

Correct!

tag

orrect Answers

tag

TAG

Tag

Question 6

1 / 1 pts

Assume a *tiny* L1 cache with a capacity of 1024 bytes and 32-byte blocks. The cache is two-way set associative. To which set does memory address 0x3a17 map?

Correct!

0

orrect Answers

0

zero

0x0

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:

```
0x1a7
0xaf1
0xcd9
0xc2a
0x782
0xac8
0xed8
0x9fe
0xf43
0x44d
0x898
0xc55
0x28c
0xbe2
0x4b3
0x747
```

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

- True
- Correct! False

Question 8

1 / 1 pts

Match the cache policies on the left with the scenario in which they may apply on the right.

- Correct! **write-allocate** — cache miss on a write
- Correct! **write-back** — cache hit on a write
- Correct! **write-through** — cache hit on a write
- Correct! **no write-allocate** — cache miss on a write

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;
```

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;
```

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;
```

```
    total_y += grid[i][j].y;
  }
```

Correct!

25

orrect Answers

25

25%

25 %

twenty five

twenty five percent

twenty-five

twenty-five percent

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];
```

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[___][___];
```

Blank 1: W

Blank 2: i

Blank 3: i+V

Blank 4: j

Blank 5: l

Blank 6: k

Blank 7: k

Blank 8: l

(NOTES: Blanks are ordered top to bottom, left to right. *Exact* 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! l

Answer 6:

Correct! k

Answer 7:

Correct! k

Answer 8:

Correct! l

Quiz Score: **14** out of 14