User ID: **KEY** CS-3160 In-Class Quiz #3 NAME: **KEY**

#1 The notion of a 'rendezvous' is associated with which synchronization method?

(a) monitors.
(b) semaphores.
● message passing.
(d) callback methods.
(e) schedulers.

#2 A task that does not communicate with or affect the execution of any other task in the program is said to be

● disjoint.
(b) singular.
(c) asynchronous.
(d) autonomous.
(b) distinct.

#3 Which levels of concurrency pose challenges and issues for programming language design?

(a) instruction and program
● statement and subprogram
(c) instruction and statement
(d) instruction and unit
(e) program and subprogram

#4 Which of the following describes an exception that might be considered normal operation.

(a) Attempting to divide by zero.
● Attempting to read past the end of a file.
(c) Attempting to read past the end of an array.
(d) Attempting to access memory outside of that assigned to the program.
(e) Exceptions are always errors.

#5 Languages that do not support exception handling

● are capable of handling software-detectable errors with user-defined code.
(b) are incapable of handling any type of exception.
(c) have not existed since the early days of computing.
(d) must provide some alternate means for handling at least hardware-detectable exceptions.
(e) do not exist since all languages provide at least some level of exception handling.

Enter the letter(s) of each answer below. You may choose multiple answers, but credit will be divided by the number of choices made.

1 **C** 2 **A** 3 **B** 4 **B** 5 **A** 6 **D** 7 **B** 8 **A** 9 **B** 10 **A**

#6 At the unit level, the exception handlers need to be dynamically bound to exceptions events, for example divide-by-zero, because

(a) exception handling is only possible in dynamically-scoped languages.
(b) exceptions are dynamic events and therefore must be dynamically bound.
(c) this is easier to implement than statically-bound exception handlers.
● different instances of the same exception often require different actions.
(e) exception handlers are called by the operating system.

#7 Allowing exceptions to propagate out of the subprogram that caused the exception to be raised

(a) is required to prevent unhandled exceptions.
● permits fewer, but more general-purpose, exception handlers to be used.
(c) is only possible in languages that permit nested subroutine definitions.
(d) violates the concept of static scoping and thus is only allowed in dynamically-scoped languages.
(e) forces a language to implement closures in order to preserve the referencing environment where the exception was raised.

#8 Once an exception is handled, the question of where to resume program execution is known as the issue of

● continuation.
(b) finalization.
(c) reintegration.
(d) resumption.
(e) termination.

#9 Being able to define code that is to be executed by a subprogram even after an exception has been handled is known as

(a) continuation.
● finalization.
(c) resumption.
(d) termination.
(e) reintegration.

#10 The main difference between event- and non-event-driven code is that

● in event-driven code the order in which code is executed is dictated by external actions rather in the code itself.
(b) non-event-driven code must rely on event-handlers to process event occurrences.
(c) event-driven code cannot be statically scoped.
(d) event-driven code must be developed in an object-oriented fashion.
(e) event-driven code is used exclusively by GUI interfaces.