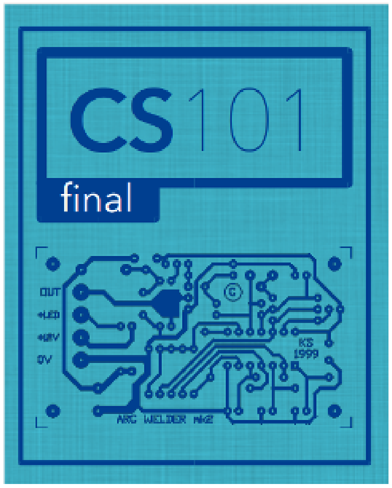


- **Be sure to enter your information below, enter your answers for multiple-choice questions on the next page, and your code for the last two coding questions on the last two pages**. Do not turn this page until instructed to.
- This is a 180-minute exam with 30 questions:
  - 12 MATLAB multiple-choice questions worth 5 points each;
  - 16 Python multiple-choice questions worth 5 points each; and
  - 2 coding questions worth 30 points each

  for a total of 200 possible points.
- **Your are NOT allowed to take any page of this final exam out of the final exam classroom. In other words, all pages of your final exam must be submitted.**
- Each multiple choice question has only *one* correct answer.
- You must not communicate with other students during the exam.
- No books, notes, or electronic devices are permitted. In other words, you are not allowed to use a dictionary on your mobile phone or other electronic devices. However, if you don't understand the meaning of a particular English word in this exam, please raise your hand and the instructor will explain the meaning of the English word to you.

1. Fill in your information:

Full Name: ______________________________

Student ID: ______________________________

The following 12 questions involve MATLAB.

1. (5 points) Consider the following MATLAB program:

```
x = [ 1 2 ];
y = [ 3 4 ];
z = [ y x ; x y ]';
```

What is the **value** of `z` after this program executes?

A. $\begin{bmatrix} 3 & 1 \\ 4 & 2 \\ 1 & 3 \\ 2 & 4 \end{bmatrix}$

B. $\begin{bmatrix} 3 & 4 & 1 & 2 \\ 1 & 2 & 3 & 4 \end{bmatrix}$

C. $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{bmatrix}$

D. $\begin{bmatrix} 1 & 3 \\ 2 & 4 \\ 3 & 1 \\ 4 & 2 \end{bmatrix}$

E. None of the other answers are correct

2. (5 points) Consider the following MATLAB program:

```
A = ones( 3,3 ) - eye( 3,3 );
A = A * 2;
A( 1:2,: ) += 3;
```

What is the **value** of `A` after this program executes?

A. $\begin{bmatrix} 3 & 5 & 5 \\ 5 & 3 & 5 \\ 2 & 2 & 0 \end{bmatrix}$

B. $\begin{bmatrix} 3 & 5 & 2 \\ 5 & 3 & 2 \\ 5 & 5 & 0 \end{bmatrix}$

C. $\begin{bmatrix} 0 & 2 & 2 \\ 5 & 3 & 5 \\ 5 & 5 & 3 \end{bmatrix}$

D. $\begin{bmatrix} 0 & 5 & 5 \\ 2 & 3 & 5 \\ 2 & 5 & 3 \end{bmatrix}$

E. None of the other answers are correct

3. (5 points) Consider the following MATLAB function stored in `squrge.m`:

```
function [ a b ] = squrge( x,y )
  a = x .^ 2;
  b = a .* 3 + y;
end
```

Which of the following correctly assigns the results of a call to `squrge` `a` to `A` and `b` to `B`, respectively?

A. `A,B = squrge( 5,4 );`

B. `[ A B ] = squrge( [ 5 4 ] );`

C. `[ A B ] = squrge( 5 4 );`

D. `[ A B ] = squrge [ 5 4 ];`

E. `[ A B ] = squrge( 5,4 );`

4. (5 points) Recollect that MATLAB represents polynomials as an array of coefficients from the highest-order coefficient to the lowest. For instance,

$$3x^2 + 2x + 1$$

is written as the array `[ 3 2 1 ]`.

How would we represent the summation of the two polynomials

$$-x^2 + 3x + 1$$

and

$$2x^3 + 4x - 1$$

as a MATLAB polynomial array?

A. `[ -1 3 1 ] + [ 2 4 -1 ]`

B. `[ -1 3 1 ] + [ 2 0 4 -1 ]`

C. `[ 0 -1 3 1 ] + [ 2 0 4 -1 ]`

D. `[ 1 3 -1 0 ] + [ -1 4 0 2 ]`

E. `[ 1 3 -1 ] + [ -1 4 2 ]`

5. (5 points) Consider the following two-dimensional MATLAB array, stored in the variable `A`:

$$\begin{bmatrix} 1 & 16 & 256 \\ 2 & 32 & 512 \\ 4 & 64 & 1024 \\ 8 & 128 & 2048 \end{bmatrix}$$

How can we index and retrieve the value 128 from this array?

A. `A( 2,4 )`

B. `A( 4,2 )`

C. `A( 1,3 )`

D. `A[ 2,4 ]`

E. `A[ 3,1 ]`

6. (5 points) *For this problem, you should compose a function which accomplishes a given task using the available code blocks arranged in the correct functional order.*

Compose a function `cross_prod` which accepts two column vectors `a` and `b` and returns the value of the cross product,

$$\vec{c} = \vec{a} \times \vec{b} = \left[a_2b_3 - a_3b_2a_3b_1 - a_1b_3a_1b_2 - a_2b_1\right].$$

```
1 end

2 c(1) = a(2)*b(3) - a(3)*b(2);

3 function [ c ] = cross_prod( a,b )

4 c(2) = a(3)*b(1) - a(1)*b(3);

5 c = zeros( 3,1 );

6 c(3) = a(1)*b(2) - a(2)*b(1);

7 c = zeros( 1,3 );

8 c = a .* b - b .* a;

9 function cross_prod( a,b )
```

A. 3, 7, 2, 4, 6, 1

B. 3, 5, 2, 4, 6, 1

C. 9, 5, 8, 1

D. 9, 7, 2, 4, 6, 1

E. 3, 7, 8, 1

7. (5 points) Consider the following MATLAB program:

```
s = (3 < 5) | ((2 > 3) & (1 ~= 0))
```

What is the final value of `s`?

A. `True`

B. `1`

C. `0`

D. `false`

8. (5 points)

```
x = eye( 2,2 );
y = [ x(2,:) ; x(1,:) ];
A = [ x y ; y x ];
```

What is the final value of `A( 2:3,2:3 )`?

A. `[ 0 1 ; 1 0 ]`

B. `[ 1 1 ; 1 1 ]`

C. `[ 0 0 ; 0 0 ]`

D. `[ 1 0 ; 0 1 ]`

9. (5 points)

```
x  = linspace( -10,10,201 );
y1 = sin( x );
y2 = cos( x );
y3 = randn( 1,numel(x) );
```

How would you successfully plot all three of these data series as points? (Assume any given plot format strings are valid.)

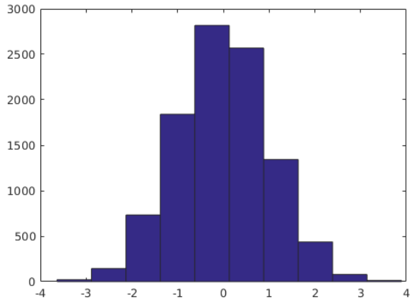
A. `plot( x, y1,'r.', y2,'g.', y3,'b.' );`

B.
```
plot( x, y1, 'r.' );
plot( x, y2, 'g.' );
plot( x, y3, 'b.' );
```

C.
```
hold on;
plot( x, y1, 'r.' );
plot( x, y2, 'g.' );
plot( x, y3, 'b.' );
```

D. `plot( x,y1, x,y2, x,y3 );`

10. (5 points) Consider the following plot, produced from 10,000 random numbers selected from an as-yet-undetermined distribution.



Which of the following MATLAB programs could produce this plot? Assume that all programs work as written.

A.
```
x = rand( 10000,1 );
plot( x );
```

B.
```
x = randn( 10000,1 );
hist( x );
```

C.
```
x = rand( 10000,1 );
hist( x );
```

D.
```
x = randn( 10000,1 );
plot( x );
```

11. (5 points)

```
A = eye( 3,3 );
for x = 1:2:3
    A( x,x ) = 0;
end
```

What is the final value of A?

A. $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

B. $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

C. $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$

D. $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

E. $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

12. (5 points)

```
A = eye( 3,3 ) - ones( 3,3 );
for x = 1:3
    for y = 1:3
        if x <= y
            A( x,y ) = x + y;
        end
    end
end
```

What is the final value of A?

A. $\begin{bmatrix} 2 & 3 & 4 \\ -1 & 2 & 5 \\ -1 & -1 & 2 \end{bmatrix}$

B. $\begin{bmatrix} 2 & 3 & 4 \\ -1 & 4 & 5 \\ -1 & -1 & 6 \end{bmatrix}$

C. $\begin{bmatrix} 2 & -1 & -1 \\ 3 & 2 & -1 \\ 4 & 5 & 2 \end{bmatrix}$

D. $\begin{bmatrix} -1 & -1 & -1 \\ 2 & -1 & -1 \\ 3 & 4 & -1 \end{bmatrix}$

E. $\begin{bmatrix} -1 & -1 & -1 \\ 3 & -1 & -1 \\ 4 & 5 & -1 \end{bmatrix}$

The following 16 questions involve Python.

13. (5 points) Consider the following incomplete Python program:

```
a = 'DWALIN'
b = 'THORIN'
d = { }
for x,y in zip( a,b ):
    ???
s = ''
for c in a:
    s += d[ c ]
```

What should replace the three question marks to cause this program to yield a final value for `s` of 'THORIN'?

A. `d[ x ] = y`

B. `d[ y ] = x`

C. `d[ a ] = b`

D. `d[ b ] = a`

E. `d[ a ] = x`

14. (5 points) Consider the following Python program:

```
d = { 0:0,1:0,2:0 }
for i in range( 10,15 ):
    d[ i%3 ] += i
x = d[ 1 ]
```

What is the final *value* of `x`?

A. `12`

B. `23`

C. `11`

D. `25`

E. `1`

15. (5 points) Consider the following Python program:

```
d = { "B":1,"A":1,"G":2,"I":1,"N":1,"S":1 }
for c in "BILBO":
    print( d[ c ] + '-' )
```

What kind of exception will this program throw?

A. `KeyError: 'L'`

B. `TypeError: list indices must be integers, not str`

C. `SyntaxError: invalid syntax`

D. `TypeError: unsupported operand type(s) for +: 'int' and 'str'`

16. (5 points) Consider the following Python program:

```
e = list( range( 0,10,2 ) )
d = [ 0,0,0,0 ]
for i in range( 0,len(e) ):
    d[ i%4 ] += e[ i ]
x = d[ 1 ]
```

What is the final *value* of `x`?

A. `0`

B. `8`

C. `10`

D. `2`

E. `14`

17. (5 points) Consider the following incomplete Python program:

```
sum = 0
???:
    sum += i
```

The program is intended to sum all of the integers between 1 and 100 (inclusive). What should replace the three question marks to complete the program?

A. `for i in range( 0,100 )`

B. `while i <= 100`

C. `for i in range( 1, 101 )`

D. `while i in range( 100 )`

18. (5 points)

```
x = np.array( [ [ 2 ] , [ 3 ] ] * 2 )
```

What is the final *value* of x?

A. $\begin{bmatrix} 2 & 2 \\ 3 & 3 \end{bmatrix}$

B. $\begin{bmatrix} 2 \\ 3 \\ 2 \\ 3 \end{bmatrix}$

C. $\begin{bmatrix} 2 & 3 & 2 & 3 \end{bmatrix}$

D. $\begin{bmatrix} 2 & 3 \\ 2 & 3 \end{bmatrix}$

19. (5 points)

```
import itertools
x = 'beorn'
???
    print( x )
```

Replacing the three question marks with which of the following will result in `'beorn'` being printed exactly five times?

A. `for a in itertools.combinations(x,5):`

B. `for a in itertools.combinations(x,2):`

C. `for a in itertools.combinations(x,3):`

D. `for a in itertools.combinations(x,4):`

20. (5 points) Consider the following incomplete Python program:

```
y = 1.0   # initial position, m
v = 0.0   # initial velocity, m/s
g = -9.8  # acceleration due to gravity, m/s^2
t = ???   # initial time, s
nt = ???  # number of time intervals, -
dt = t/nt # time increment, s

while y > 0.0:
    t += dt
    v += g * dt
    y += v * dt
```

Which of the following values for `t` and `nt` will yield the most accurate solution?

A. `t,nt = 1.0,1e5`

B. `t,nt = 10.0,1e3`

C. `t,nt = 10.0,1e4`

D. `t,nt = 1.0,10`

21. (5 points)

```
s = 'THRANDUIL'
x = ''
for i in range( 0,len( s ) ):
    if ( i>3 ) and ( i<6 ):
        x += s[ i:i+2 ]
```

What is the *value* of `x` after this program is executed?

A. `'NDDU'`

B. `'ANND'`

C. `'AN'`

D. `'ND'`

E. None of the other answers are correct.

22. (5 points)

```
def sum_pairs( A ):
    total = 0
    ???
    return total
```

The function `sum_pairs` accepts a list of floats named `A`. `sum_pairs` should return the sum of all pairs of values in the list (without repeats). For example, given the list `[ 1,2,3 ]`, `sum_pairs` should return `12` from $(1+2)+(1+3)+(2+3)=12$. What should replace the three question marks to complete the function? (Assume any necessary `import`s to have taken place already.)

A.
```
for i in range( len( A ) ):
    for j in range( i+1,len( A ) ):
        total += A[ i ] + A[ j ]
```

B.
```
for i in range( len( A ) ):
    for j in range( len( A ) ):
        total += A[ i ] + A[ j ]
```

C.
```
for i,j in enumerate( A ):
        total += A[ i ] + A[ j ]
```

D.
```
for i in itertools.permutations( A ):
        total += i[ 0 ] + i[ 1 ]
```

23. (5 points) What do we call the optimization heuristic that involves iteratively checking to see if neighboring solutions improve upon the current solution?

A. Conjugate gradient

B. Local optimum

C. Hill-climbing

D. Random search

24. (5 points)

```
def total_sales( sales_file ):
    d = { }
    for line in open( sales_file ):
        ???
    return d
```

The function `total_sales` should compute the total sales of each employee working for a company by reading a comma-separated value input file of employee sale data. The result should be returned from the function as a dictionary. The first column of each line in the input file is expected to contain the employee's name represented as a string. The second column is expected to contain a floating point number representing the total for that sale. Here is an example input file:

```
Tom,10.0
Bill,10.55
Bill,115.50
Your program should ignore a non-conforming line like this one.
Bert,30.25
```

The resulting return value for this file should be the following dictionary:

```
{ 'Bert':30.25, 'Bill':126.05, 'Tom':10.0 }
```

What should replace the three question marks to complete the function?

A.
```
try:
    s,f = line.split( "," )
    if s not in d:
        d[ s ] = 0.0
    d[ s ] += float( f )
except:
    continue
```

B.
```
if line not in d:
    d[ line ] = 0.0
try:
    s,f = line.split( "," )
except:
    d[ s ] += float( f )
    continue
```

C.
```
try:
    s,f = line.split( "," )
except:
    continue
if f not in d:
    d[ f ] = 0.0
d[ f ] += float( s )
```

D.
```
  try:
      s,f = line.split()
      d[ s ] += float( f )
  except:
      break
```

25. (5 points)

```
s = ''.join( [ "0","1","2","1" ] )
x = 0
for i in range( len( s )-1 ):
    x += int( ??? )
```

What should replace the three question marks so the resulting value of x is 34?

A. `s[ i:i+2:i ]`

B. `s[ i:i+1 ]`

C. `s[ i+2:i:-1 ]`

D. `s[ i+1:i+2 ]`

26. (5 points)

```
x = [ ]
for i in range( 1,101 ):
    for j in range( i+1,101 ):
        t = i,j
        x.append( t )
```

After the program runs, which of the following is an element of `x`?

A. `(10,52)`

B. `(0,33)`

C. `(42,15)`

D. `(78,78)`

E. `(11,4)`

27. (5 points)

```
e = [ 1,1,2,2,3,3,4,4,5,5 ]
d = { 0:0,1:0,2:0 }
for a,b in enumerate( e ):
    d[ a%3 ] += b
x = d[ 1 ]
```

After it is run, what is the final *value* of `x`?

A. `3`

B. `10`

C. `12`

D. `22`

E. `8`

28. (5 points)

```
x = "5 4 1".split()
x = x.sort()
try:
    print( len( x ) )
except:
    print( type( x ) )
```

After it is run, what is printed by this program?

A. `TypeError`

B. `3`

C. `list`

D. `NoneType`

29. (25 points)

You have been hired by a private investigation firm to crack an smartphone of indeterminate provenance (and a process of questionable legality). The default password is exactly five characters long, with possible characters selected from the upper- and lower-case alphabets and the ten digits `0` to `9`. Assume that you have available a function `test_password` which returns `True` if the password is correct and `False` otherwise.

Compose a Python function `crack_phone` which accepts no arguments and returns a string representing the correct password which unlocks the smartphone. You may `import itertools` in your solution if you prefer, but no other libraries are allowed.

```
alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789'
```

30. (25 points) Consider the Taylor series definition of the sine function:

$$\sin(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + ...$$

The series converges for all real x, so to calculate $\sin(x)$ to within a few decimal places of accuracy one just needs to include sufficient terms in the calculation.

The following MATLAB function `sine` was written in order to calculate the value of $\sin(x)$ for all x to three decimal places of accuracy (`atol` in the code). Translate this function into a Python function—also called `sine`—which yields identical output from the function as the MATLAB function for given input. You may `import numpy as np` in your solution if you prefer, but no other libraries are allowed. (Assume a valid NumPy-compatible function `factorial` is also available.)

```
function [ y ] = sine( x )
    y = 0;
    yold = 1;
    n = 0;
    atol = 1e-3;  % tolerance
    while ( abs( y-yold ) > atol )
        yold = y;
        term = ( x .^ ( 2*n+1 ) ) / factorial( ( 2*n+1 ) );
        if (mod(n,2) == 1)
            term = -term;
        end
        y = y + term;
        n = n + 1;
    end
end
```