Instructions:

1. Answer all questions on this paper. For multiple choice questions circle the letter of the **best** choice. For short answer questions, write your answer in the space provided.
2. No aids (such as calculators or language translators) are permitted.
3. Do NOT remove any pages from the exam.
4. You have 3 hours to complete the exam. There is a total of 50 marks.

Part 1: Predict the output [12 x 1 mark]

In each row of the table below, mentally execute the code fragment on the left and enter the expected output in the box on the right. Assume any necessary imports have been done. Each row is separate.

	Code Fragment	*Expected Output*
1.	`print(16 / -2**3 * 2)`	
2.	`print(15 % 10 // 2 * 5)`	
3.	`ar = np.array([1,2,3,4,5,6])` `print(len(ar.reshape(3,2)))`	
4.	`print(np.linspace(4,-2,4))`	
5.	`ss = "{abc}{&*@}{123}"` `print(ss.split('}{'))`	

6.	`ss = {1,2,3}`
	`ss.add(2)`
	`ss.add(4)`
	`print(ss)`	

7.	`print('ot,'.join(['h','p','n']))`	

8.	`dd ={1:5,4:2}`
	`for xx in dd:`
	`  print(xx,dd[xx])`	

9.	`print(np.array([2,5])**2 +` `  np.arange(-2,0))`	

10.	`ar = np.arange(4)`
	`ar[2], ar[3] = ar[3], ar[2]`
	`print(ar)`	

11.	`ii = np.array([1,2])`
	`jj = np.array([3,1])`
	`print(np.dot(ii,jj))`	

12.	`print(np.zeros((2,3),bool))`	

Part 2: Circle the letter of the best answer. [8 x 1 mark]

13. With respect to random numbers created by the functions available in Python which of the following statements is **true**?

a) The random numbers are truly random.
b) The random numbers are always positive.
c) The random numbers are always between 0 and 1.
d) The sequence of random numbers can be made the same every time a program is run.
e) The random numbers are always floats.

14. With respect to arrays which of the following statements is **false**?

a) All the elements of the array must be of the same type.
b) Vector arithmetic can be used with arrays of the same shape.
c) The number of elements in an array cannot be changed.
d) The shape of an array can be changed.
e) A slice of an array is a new array.

15. Which of the following is NOT a valid string literal?

a) 'Computers don't think!'
b) "Computers don't think!"
c) 'Computers don\'t think!'
d) "Computers don\'t think!"
e) "Computers don\"t think!"

16. Which of the following is statements is **true**?

a) The data value of an element of a dictionary cannot be another dictionary.
b) The data value of an element of a dictionary cannot be a set.
c) The name of a dictionary can be used in a for statement.
d) The elements in a dictionary are in ascending order of the values of the keys.
e) The keys of a dictionary must be strings.

17. Which of the following statements is t**rue**?

a) Every function definition must end with a return statement.
b) Every if statement must must have an else clause.
c) Every function name must be followed by (.
d) Every loop must contain a break statement.
e) Every tuple literal must be enclosed in parentheses.

18. For which of the following can the contents **not** be changed after it has been created?

a) list
b) tuple
c) set
d) dictionary.
e) array

19. What is the type of the shape of an array?

a) int
b) float
c) numeric
d) list
e) tuple

20. Which of the following statements is **false**?

a) The * operator can be used to replicate a string.
b) The * operator can be used to replicate a tuple.
c) The * operator can be used to replicate a list.
d) The * operator can be used replicate a set.
e) The * operator can be used to perform multiplication of numeric values.

Use this area for scrap work.

Part 3: Programming [Total 30 marks]

21. **[5]** Complete the Python function that computes the approximate volume of a cone using the ~~Monte Carlo technique~~. The function is given the *radius* of the base of the cone, the *height* of the cone and the number of random *points* to create. The cone is fully enclosed in a box whose X and Y coordinates range from -*radius* to *radius* and whose Z coordinates range from 0 to *height*. Create three arrays, *xCoords*, *yCoords* and *zCoords* to hold the X, Y and Z coordinates of the random points inside the box. Use the arrays and vector arithmetic to compute the probability of a point being inside the cone. A point (x,y,z) is inside the cone if $x^2+y^2 \le r^2$, where $r=radius(height-z)/height$. Compute the approximate volume of the cone using the probability and the volume of the box enclosing the cone. Return the probability followed by the approximate volume.

Note: You must use *vector arithmetic*. There must not be any loops in this function.

Write the import statement(s) required by the function here.

```
def monteCone(radius, height, points):
```

22. Complete the Python function that begins with the header given below. The parameter *filename* is the name of the file to be read. The first few lines of the file might be:

[6]

```
00:00:00   01:12:59  06:45:31
10:10:10
13:21:17 15:19:31
```

Each line consists of one or more times of the form hh:mm:ss, where hh is the hour, mm is the minute and ss is the second. The hour, minute and second are separated from each other by a :. The times are separated from each other by whitespace. There may be more than one space separating the times on a line. Read the lines from the file and create lists of **integer** hours, **integer** minutes and **integer** seconds from the data in the file. Return the lists of hours, minutes and seconds in that order.

```
def readFile(fileName):
```

23. Complete the Python function that begins with the header given below. This function computes the four equations of motion of an object under constant acceleration, as given below.

[4]

$$v_{f0} = v_0 + a \cdot t \qquad (equation\ 1)$$

$$x_{f0} = x_0 + v_0 \cdot t + \frac{a \cdot t^2}{2} \qquad (equation\ 2)$$

$$v_{f1} = \sqrt{v_0^2 + 2 \cdot a (x_{f0} - x_0)} \qquad (equation\ 3)$$

$$x_{f1} = x_0 + \frac{(v_{f0} + v_0) t}{2} \qquad (equation\ 4)$$

Equations 1 and 3 compute the final velocity of the object in two different ways. Equations 2 and 4 compute the final position of the object in two different ways. In the equations x_0 is the initial position of the object, v_0 is the initial velocity of the object, a is the constant acceleration of the object and t is the current time value. Create an array of equally spaced time values from t=0 to t=duration based on the number of intervals. Use equation 1 and vector arithmetic to create an array of final velocities named vf0. Use equation 2 and vector arithmetic to create an array of final positions named xf0. Use equation 3 and vector arithmetic to create a second array of final velocities named vf1. Use equation 4 and vector arithmetic to create a second array of final positions named xf1. There must not be any loops in this function. Return the arrays vf0, vf1, xf0, xf1, in that order.

Write the imports statement(s) required by the function here.

```
def motionEquations(x0, v0, a, duration, intervals):
```

24. Complete the Python function that begins with the header given below. The function is given a dictionary named *products*. The key of a dictionary element is the name of the product as a string.
[3] The data value of a dictionary element is a tuple, where the first element of the tuple is the cost of manufacturing the product as a float, the second element of the tuple is the wholesale price of the product as a float and the third element of the tuple is the retail price of the product as a float. Display a heading as shown in the sample output given below. There must be a blank line before the heading. For each product display the product name left justified in 15 character positions, the cost, wholesale price and retail price in 10 character positions to 2 decimal places on one line. Separate the values displayed by one space. This function is used in question 25.

Sample output from this function might be:

```
Product              Cost  Wholesale     Retail
shoe                29.99      45.88      56.00
hat                 17.00      24.22      32.38
glove               14.49      20.24      27.21
```

```
def displayPrices(products):
```

25. **[7]** Write a Python main program that calls *displayPrices* from question 24. Do NOT rewrite *displayPrices* here. There is no function named *main*. The program creates a dictionary of products, where the key of an element in the dictionary is a product name as a string. The data value of an element is a tuple of three floats: the cost of manufacturing the product, the wholesale price of the product and the retail price of the product. In the program do the following:

- Display a line of 80 dashes using string replication. There must be a blank line before the dashes.
- Input the name of a product as a string and remove the whitespace from the beginning and end of the string.
- Loop as long as the name is not an empty string.
- In the loop:
  - If the product is in the dictionary display the message 'Duplicate product: ' followed by the name of the product as shown in the sample run of the program. Otherwise do the following.
  - Input the cost of the product as a float.
  - If the cost is less than or equal to zero display an error message that includes the cost as shown in the sample run of the program. Otherwise do the following.
  - Compute the wholesale price of the product using equation 5.
  - Compute the retail price of the product using equation 6.
  - Add an element to the dictionary of products where the product name is the key and the data value is a tuple containing the cost, wholesale price and retail price in that order.
  - Input the name of a product as a string and remove the whitespace from the beginning and end of the string.
- Call *displayPrices* to display the dictionary of products.

$$wholesale = (\sqrt{cost})^{2.25} \qquad (equation 5)$$

$$retail = (\log_e(wholesale))^3 \qquad (equation 6)$$

Be sure to include any import statements required by the program.

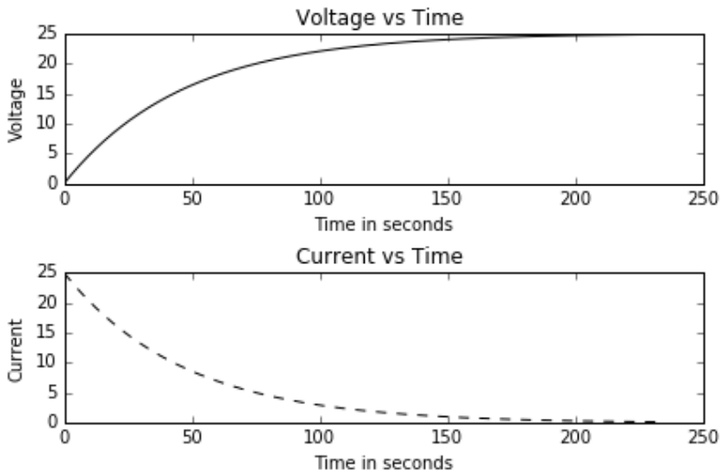
A sample run of the program is shown below.

```
--------------------------------------------------------------------------

Enter the name of the product: hat

Enter the cost of the product (>0): 19.99

Enter the name of the product: hat
Duplicate product: hat

Enter the name of the product: jacket

Enter the cost of the product (>0): -79.95
The cost of the product, -79.95, must be greater than zero!

Enter the name of the product: jacket

Enter the cost of the product (>0): 79.95

Enter the name of the product: boots

Enter the cost of the product (>0): 109.49

Enter the name of the product:

Product               Cost  Wholesale     Retail
hat                  19.99      29.07      38.26
jacket               79.95     138.25     119.76
boots               109.49     196.92     147.43

Programmed by Stew Dent.
Date: Wed Jul 18 15:21:44 2018
End of processing.
```

Write the main program on the next page.

Use this page for the solution to question 25.
Question 26 is on the next page.

26. **[5]** Complete the Python function that begins with the header given below to display two subplots in a single figure. Use functions from *pyplot* to create the plots. Specify black as the colour of the curves to plot. The first curve is a solid line, the second curve is a dashed line. Use the arrays *times* and *voltages* for the curve in the first subplot and the arrays *times* and *currents* for the curve in the second subplot. Use *title1* to label the y-axis and as part of the title for the first curve and *title2* to label the y-axis and as part of the title for the second curve. Label the x-axis as shown below. Use *title1* and *title2* to create the name of the file to save. Save the figure as a .png file. Be sure to display the plot.



Write the import statement(s) required by the function here.

```
def plotCurves(times, voltages, currents, title1, title2):
```

Use this page to complete the answer to any of the questions or for scrap work. Do **NOT** remove this page from the exam.

Use this page to complete the answer to any of the questions or for scrap work. Do **NOT** remove this page from the exam.