Analysis with Programming Tools (ITP4869)

Written Test

Marking Scheme

Q1 (a)	✓ High-level ✓ Interactive ✓ Object-oriented ✓ Scripting ✓ Code readability ✓ Functional ✓ Procedural ✓ General purpose ✓ Dynamic type system ✓ Automatic memory management <Any three of the above, 1 mark each>	3
(b)	`a, b = b, a`	2

(c)	`import math`
	`def area(r):`
	`  return r * r * math.pi`
		1
		1
		3

	Total	**10**

Q2 (a)	`n = int(input("Please enter an integer: "))`
	
	`print([n**2, (n+1)**2, (n+2)**2])`
	`# Or`
	`print([(n+i)**2 for i in range(3)])`
		2
		
		3

(b)	`jargon = ['python', 'spark', 'hadoop', 'rdd']`
	`jargon.sort()`
	`jargon.reverse()`
	`jargon`
	`# Or`
	`jargon = ['python', 'spark', 'hadoop', 'rdd']`
	`jargon2 = sorted(jargon)[::-1]`
	`jargon2`
		
		2
		2
		1
		Or
		
		4
		1

	Total	**10**

Q3	`class Rectangle():`
	`  def __init__(self, l, w):`
	`  self.length = l`
	`  self.width  = w`
	
	`  def area(self):`
	`  return self.length * self.width`
	
	`  def perimeter(self):`
	`  return 2 * (self.length + self.width)`
		1
		1
		1
		1
		
		1
		2
		
		1
		2

	Total	**10**

Q4 (a)	`narr = np.arange(11,23).reshape(3,4)`	3
(b)	`narr2 = narr[:,-1]`	2
(c)	`narr3 = narr[narr%3 == 0]`	2

(d)	`narr4 = narr2.T ** 2`
	or
	`narr4 = np.square(narr2.T)`
		3

	Total	**10**

Q5 (a)	`df = pd.read_csv('scores.csv', index_col='SID')`	3
(b)	`df.loc[[18002,18005]]`	2
(c)	`df[df['Python']>80]`	2
(d)	`df['Total'] = df.sum(axis=1)`	3
	Total	**10**

Q6 (a)	`df2 = pd.concat([s1, s2, s3], axis=1, keys=['cat1', 'cat2', 'cat3'])`	3
(b)	`df3 = df2.replace(np.nan, 'X')`	2
(c)	`df4 = df3.loc['c':'f']`	2
(d)	`df4.columns = df4.columns.map(str.title)`	3
	Total	**10**

***** End of Marking Scheme *****