PES University, Bengaluru

(Established under Karnataka Act No. 16 of 2013)

UE18CS306B

OCTOBER 2020: IN SEMESTER ASSESSMENT B Tech 5 SEMESTER

TEST – 2

UE18CS306B (2 credit subject) - Python Application Programming

Time: 40 min	Answer All Questions	Max Marks: 20

1.	a)	Write a python program to create cursor object
		
		import mysql.connector
		
		#establishing the connection
		conn = mysql.connector.connect(
		user='root', password='password', host='127.0.0.1', database='mydb'
		)
		#Creating a cursor object using the cursor() method
		cursor = conn.cursor()
			2

	b)	Describe the Engine class and explain any 5 methods of Engine class Engine class connects a **Pool and Dialect together** to provide a source of database **connectivity and behavior**. An object of Engine class is instantiated using the **create_engine()** function from sqlalchemy import create_engine engine = create_engine('sqlite:///college.db', echo = True 1)**connect()** Returns connection object **2) execute()** Executes a SQL statement construct **3) begin()** Returns a context manager delivering a Connection with a Transaction established. Upon successful operation, the Transaction is committed, else it is rolled back **4) dispose()** Disposes of the connection pool used by the Engine **5) driver()** Driver name of the Dialect in use by the Engine	3
	c)	Write a python program that demonstrates the use of the following Sqlite3 module routines, with suitable comments. a) sqlite3.connect() b) connection.cursor() This routine creates a **cursor** which will be used throughout database programming with Python. This method accepts a single optional parameter cursorClass. c) cursor.execute() This routine executes an SQL statement. The SQL statement may be parameterized (i. e. placeholders instead of SQL literals). The sqlite3 module supports two kinds of	5

placeholders: question marks and named placeholders (named style)

For example − cursor.execute("insert into people values (?, ?)", (who, age))

d) connection.commit()

This method commits the current transaction. If this method is not called, anything did since the last call to commit() is not visible from other database connections.

e) connecton.rollback()

This method rolls back any changes to the database since the last call to commit().

2.	a)	Write a server side connection oriented socket program that binds local IP address and port number 60 to receive data of size 30bytes and print to the console
		
		import socket
		LOCALHOST = "127.0.0.1"
		PORT = 60
		server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
		server.bind((LOCALHOST, PORT))
		server.listen(1)
		print("Server started")
		print("Waiting for client request..")
		clientConnection,clientAddress = server.accept()
		print("Connected clinet :" , clientAddress)
		msg = ''
		while True:
		  in_data = clientConnection.recv(30)
		  msg = in_data.decode()
		  if msg=='bye':
		  break
		  print("From Client :" , msg)
		  out_data = input()
		  clientConnection.send(bytes(out_data,'UTF-8'))
		print("Client disconnected....")
		clientConnection.close()
			5

	b)	Write a client side UDP program.
		
		import socket
		# For creating the udp socket
		udp_socket = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
		# Host IP
		udp_host = socket.gethostname()
		# We are specifying port to connect
		udp_port = 12345
		
		msg = "Welcome to Python"
		print("UDP target IP:", udp_host)
		print("UDP target Port:", udp_port)
		# Sending message to UDP server
		udp_socket.sendto(msg,(udp_host,udp_port))
			2

	c)	List and explain any 8 general socket methods. • **socket.recv(bufsize)** − As name implies, this method receives the TCP message from socket. The argument bufsize stands for buffer size and defines the maximum data this method can receive at any one time.	3

- **socket.send(bytes)** − This method is used to send data to the socket which is connected to the remote machine. The argument bytes will gives the number of bytes sent to the socket.
- **socket.recvfrom(data, address)** − This method receives data from the socket. Two pair (data, address) value is returned by this method. Data defines the received data and address specifies the address of socket sending the data.
- **socket.sendto(data, address)** − As name implies, this method is used to send data from the socket. Two pair (data, address) value is returned by this method. Data defines the number of bytes sent and address specifies the address of the remote machine.
- **socket.close()** − This method will close the socket.
- **socket.gethostname()** − This method will return the name of the host.
- **socket.sendall(data)** − This method sends all the data to the socket which is connected to a remote machine. It will carelessly transfers the data until an error occurs and if it happens then it uses socket.close() method to close the socket.
- **socket.bind()** − This method binds the address (hostname, port number) to the socket.
- **socket.listen()** − This method basically listens to the connections made to the socket. It starts TCP listener. Backlog is an argument of this method which specifies the maximum number of queued connections. Its minimum value is 0 and maximum value is 5.
- **socket.accept()** − This will accept TCP client connection. The pair (conn, address) is the return value pair of this method. Here, conn is a new socket object used to send and receive data on the connection and address is the address bound to the socket. Before using this method, the socket.bind() and socket.listen() method must be used.