

Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065384373
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 93 Question Id : 640653587000 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0


Question Label : Multiple Choice Question


THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MODERN APPLICATION DEVELOPMENT I (COMPUTER BASED EXAM) "

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?  
CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406531958668.  YES

6406531958669.  NO

Sub-Section Number :	2
Sub-Section Id :	64065384374
Question Shuffling Allowed :	Yes
Is Section Default? :	null

**Question Number : 94 Question Id : 640653587001 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 2**

Question Label : Multiple Choice Question

Consider the following flask application and select the correct option if the application is running locally on `http://127.0.0.1:5000`.

```
from flask import Flask

app = Flask(__name__)

@app.route('/work')
@app.route('/home')
def my_task():
    return "<h1>Hello! Reporting for my task</h1>"

app.run()
```

**Options :**

- 6406531958670. The application will return `200` OK status code for `/home` endpoint only
- 6406531958671. The application will return `200` OK status code for `/work` endpoint only
- 6406531958672. The application will return `200` OK status code for both the endpoints
- 6406531958673. The application will throw an `AssertionError` for both the endpoints

**Question Number : 95 Question Id : 640653587002 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 2**

Question Label : Multiple Choice Question

Which of the following statements about OpenAPI Specification (OAS) is true?

**Options :**

6406531958674. OAS is a programming language used for building APIs.
6406531958675. OAS is a way to describe interfaces for building RESTful APIs.
6406531958676. OAS is a tool used for testing the performance of APIs.
6406531958677. OAS is a software library used for authenticating API requests.

**Question Number : 96 Question Id : 640653587005 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 2**

Question Label : Multiple Choice Question

Map the terms in column A with their correct interpretations in column B.

	Column A		Column B
a	Scale up	1	financial transactions
b	ACID	2	adding more physical servers
c	Scale out	3	increasing RAM
d	BASE	4	social media

**Options :**

6406531958686.  $a \rightarrow 3, b \rightarrow 1, c \rightarrow 2, d \rightarrow 4$
6406531958687.  $a \rightarrow 4, b \rightarrow 2, c \rightarrow 1, d \rightarrow 3$
6406531958688.  $a \rightarrow 4, b \rightarrow 3, c \rightarrow 1, d \rightarrow 2$
6406531958689.  $a \rightarrow 3, b \rightarrow 2, c \rightarrow 1, d \rightarrow 4$

**Question Number : 97 Question Id : 640653587011 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 2**

Question Label : Multiple Choice Question

Consider the following view function.

```
@app.route('/student', methods = ['GET', 'POST'])
def show_student():
    std = request.args
    details = {
        'Department': std['dept'],
        'Course-level': std['level'],
        'Course': std['course']
    }
    return details
```

If this flask app is running locally on <http://127.0.0.1:5000>, which of the following URLs will be handled by the controller correctly?

**Options :**

6406531958710.

<http://127.0.0.1:5000/student/Electrical/Diploma/MAD1>

6406531958711.

<http://127.0.0.1:5000/Electrical/Diploma/MAD1>

6406531958712.

<http://127.0.0.1:5000/student?dept=Electrical&level=Diploma&course=MAD1>

6406531958713.

<http://127.0.0.1:5000?endpoint=student&dept=Electrical&level=Diploma&course=MAD1>

**Sub-Section Number :**

3

**Sub-Section Id :**

64065384375

**Question Shuffling Allowed :**

Yes

**Is Section Default? :**

null

**Question Number : 98 Question Id : 640653587003 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

What will be the output of the following Python code snippet on the terminal?

```
def modify(n):
    def modifier(n):
        ser = [0,1]
        for i in range(n-2):
            new = ser[i]+ser[i+1]
            ser.append(new)
        print(ser)
    return modifier

@modify
def list_num(n):
    nums = []
    for i in range(n):
        nums.append(i+1)
    print(nums)

list_num(10)
```

**Options :**

6406531958678. [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

6406531958679. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

6406531958680. [0, 1, 1, 2, 3, 5, 8, 13, 21]

6406531958681. [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

**Question Number : 99 Question Id : 640653587006 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Consider the following HTML Document file given below.

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Document</title>
  <style>
    input:invalid {
      background: red;
    }
    input:valid {
      background: green;
    }
  </style>
</head>
<body>
  <form>
    <label for="uname">Enter a valid e-mail:</label>
    <input type="text" name="uname" minlength="8">
  </form>
</body>
</html>
```

Suppose the index.html is rendered on the browser. What will be the background color of the input box when the user enters the name "madcourse.mail.com"?

**Options :**

6406531958690.    red

6406531958691.    white

6406531958692.    green



6406531958693. insufficient information

**Question Number : 100 Question Id : 640653587008 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Consider the following Python code snippet.

```
from flask import Flask
from flask_restful import Api, Resource

app = Flask(__name__)
api = Api(app)

class MyApi(Resource):
    def get(self):
        return {"greet": "Hello from GET Api!"}

    def put(self):
        return {"greet": "Hello from PUT Api!"}

api.add_resource(MyApi, '/api/get', '/api/put', '/api/post')

app.run()
```

If this application is running locally on `http://127.0.0.1:5000`, which of the following curl commands will throw an error?

1. `curl http://127.0.0.1:5000/api/get -X get`
2. `curl http://127.0.0.1:5000/api/put -X put`
3. `curl http://127.0.0.1:5000/api/post -X post`
4. `curl http://127.0.0.1:5000/api/get -X put`
5. `curl http://127.0.0.1:5000/api/put -X get`
6. `curl http://127.0.0.1:5000/api/post -X get`

**Options :**

6406531958698. Only 3

6406531958699. Only 3 and 4

6406531958700. Only 5 and 6

6406531958701. Only 3, 4, 5 and 6

**Question Number : 101 Question Id : 640653587010 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

Consider the following code.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def generate1():
    return "This is generate1"

@app.route('/')
def generate2():
    return "This is generate2"

@app.errorhandler(404)
def page_not_found(e):
    # setting 404 status explicitly
    return 'page not found'

app.run()
```

If the flask application is running on <http://127.0.0.1:5000>, what will browser render for URL <http://127.0.0.1:5000/>

**Options :**

6406531958706. Page not found

6406531958707. This is generate2

6406531958708. This is generate1

6406531958709. Code will throw error



**Question Number : 102 Question Id : 640653587016 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

Question Label : Multiple Choice Question

In the code snippet given below, what should come in place of *code 1* and *code 2* such that one parent can have multiple children and the converse does not hold true?

```
from sqlalchemy import ForeignKey
from sqlalchemy import Integer, Column
from sqlalchemy.orm import DeclarativeBase
from sqlalchemy.orm import relationship

class Base(DeclarativeBase):
    pass

class Parent(Base):
    __tablename__ = "parent_table"
    id = Column(Integer, primary_key=True)
    # write your code 1 here

class Child(Base):
    __tablename__ = "child_table"
    id = Column(Integer, primary_key=True)
    # write your code 2 here
```

**Options :**

```
code 1: parent_id=Column(Integer, ForeignKey("child_table.id"))
code 2: children = relationship("Parent")
```

6406531958727.

```
code 1: children = relationship("Child")
code 2: parent_id=Column(Integer, ForeignKey("parent_table.id"))
```

6406531958728.

```
code 1: parent_id=Column(Integer, ForeignKey("parent_table.id"))
code 2: children = relationship("Child")
```

6406531958729.

```
code 1: children = relationship("Parent")
code 2: parent_id=Column(Integer, ForeignKey("child_table.id"))
```

6406531958730.

Sub-Section Number :	4
Sub-Section Id :	64065384376
Question Shuffling Allowed :	Yes
Is Section Default? :	null

**Question Number : 103 Question Id : 640653587004 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5**

Question Label : Multiple Choice Question

Consider the following models Brand and Cellphone corresponding to tables brand and cellphone in SQLite database.

```
class Brand(db.Model):
    id = db.Column(db.Integer(), primary_key = True)
    name = db.Column(db.String(), unique = True)

class Cellphone(db.Model):
    id = db.Column(db.Integer(), primary_key = True)
    name = db.Column(db.String(), unique = True)
    brand = db.Column(db.Integer(), unique = True, db.ForeignKey("brand.id"))
```

Based on the model schemas, what relationship do the classes Brand and Cellphone share?

**Options :**

- 6406531958682. One-to-One
- 6406531958683. One-to-Many
- 6406531958684. Many-to-Many
- 6406531958685. The tables are not at all related

**Question Number : 104 Question Id : 640653587012 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5**

Question Label : Multiple Choice Question

Consider the schema for the Class Student.

```
CREATE TABLE "student" (  
    "s_id"      INTEGER,  
    "roll_number" TEXT NOT NULL UNIQUE,  
    "first_name" TEXT NOT NULL,  
    "last_name" TEXT NOT NULL,  
    PRIMARY KEY("s_id" AUTOINCREMENT)  
);
```

What will be the output of the flask\_sqlalchemy command given below?

```
>>> s1 = Student(roll_number = M01, first_name = "Yash", last_name = "Raj")  
>>> db.session.add(s1)  
>>> s2 = Student(roll_number = M02, first_name = "Yash", last_name = "Maurya")  
>>> db.session.add(s2)  
>>> s3 = Student(roll_number = M03, first_name = "Ansh", last_name = "Raj")  
>>> db.session.add(s3)  
>>> db.session.commit()  
>>> user1= Student.query.filter_by(first_name="Yash").first()  
>>> user1.first_name= "Ansh"  
>>> user1.last_name= "Maurya"  
>>> db.session.commit()  
>>> s1 = Student.query.all()  
>>> for student in s1:  
    print(student.first_name)  
    print(student.last_name)
```

**Options :**

```
Yash  
Raj  
Yash  
Maurya  
Ansh  
Raj
```

6406531958714.

6406531958715.

Ansh  
Maurya  
Yash  
Maurya  
Ansh  
Raj

Ansh  
Maurya  
Ansh  
Maurya  
Ansh  
Raj

6406531958716.

None

6406531958717.

<b>Sub-Section Number :</b>	5
<b>Sub-Section Id :</b>	64065384377
<b>Question Shuffling Allowed :</b>	Yes
<b>Is Section Default? :</b>	null

**Question Number : 105 Question Id : 640653587007 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5 Max. Selectable Options : 0**

Question Label : Multiple Select Question



Consider the following route in the flask for a signup page and select the correct option.

```
@app.route('/signup', methods=['GET', 'POST'])
def signup():
    if request.method == 'GET':
        return """<form action='/signup' method='POST'>
            <label for='username'>Username</label>
            <input type='text' name='username' required>
            <label for='password'>Password</label>
            <input type='text' name='password' required minlength="8">
            <input type='submit' value='Submit'>
            </form>
        """
    if request.method == 'POST':
        if request.form.get('username') is None:
            return redirect(url_for(signup))
        if request.form.get('password') is None:
            return redirect(url_for(signup))
        if len(request.form.get('password')) < 8:
            return redirect(url_for(signup))
        return f"<h1>Welcome, {request.form.get('username')}!</h1>"
```

### Options :

- 6406531958694. The signup page is dynamically generated.
- 6406531958695. The signup page uses server-side rendering.
- 6406531958696. The signup page uses frontend validation only.
- 6406531958697. The signup page uses backend validation.

**Question Number : 106 Question Id : 640653587009 Question Type : MSQ Is Question**

**Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4.5 Max. Selectable Options : 0**

**Question Label : Multiple Select Question**



Consider the following flask app. Given that `test_request_context()` allows text to be printed on the terminal, which of the following statements is/are correct?

```
from flask import Flask, url_for

app = Flask(__name__)

@app.route('/home')
def index():
    return 'Mad-I welcomes you!'

@app.route('/user/<username>')
def profile(username):
    return f'{username}\s profile'

with app.test_request_context():
    #== print statement ==#
```

**Options :**

6406531958702. If `#== print statement ==#` is replaced by `print(url_for('home'))`, the output on the terminal will be Mad-I welcomes you!.
6406531958703. If `#== print statement ==#` is replaced by `print(url_for('profile', username='Harry'))`, the output on the terminal will be /user/Harry
6406531958704. If `#== print statement ==#` is replaced by `print(url_for('profile', username='Harry', next='course'))`, the output on the terminal will be /user/Harry/course
6406531958705. If `#== print statement ==#` is replaced by `print(url_for('index', username='Harry'))`, the output on the terminal will be /home?username=Harry

**Sub-Section Number :**

6

**Sub-Section Id :**

64065384378

**Question Shuffling Allowed :**

Yes

Is Section Default? :

null

Question Number : 107 Question Id : 640653587013 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following is/are valid JSON format.

Options :

```
{
  "Age": 27,
  "firstName": "John",
  "lastName": "Pollard",
  "married": false,
  "phone_numbers": ["212-555-1234", "212-666-5678"]
}
```

6406531958718.

```
{
  "RollNumber": 11,
  "firstName": "Nick",
  "lastName": "Paul",
  "phone_numbers": [2125551234, "2124441234"]
}
```

6406531958719.

```
{
  "firstName": "Will",
  "lastName": "Smith",
  "address": [
    "addressLine": "Lake Union Hill Way",
    "city": "Atlanta",
    "zipCode": 50005
  ]
}
```

6406531958720.

6406531958721.

```
{
  "firstName": "Hannah",
  "lastName": "Smith",
  "cities": ["Dallas", "San Antonio", "Irving"],
  "phone_numbers": ["212-555-1234", "212-666-5678"]
}
```

**Question Number : 108 Question Id : 640653587015 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3 Max. Selectable Options : 0**

Question Label : Multiple Select Question

Consider the following flask application.

```
from flask import Flask, render_template, request
app = Flask(__name__)

@app.route('/')
def out():
    val = request.args

    if val['num'] == '':
        return "<html><body> <h1>Invalid number</h1></body></html>"
    elif val['num'].isalpha()==True:
        return "<html><body><h1>Enter a valid number</h1></body></html>"
    else:
        out = int(val['num']) * int(val['num'])
        return f'<html><body> <h1>The output is {out}</h1></body></html>'

if(__name__ == "__main__"):
    app.run(debug=True)
```

If this flask app is running locally on <http://localhost:5000>, then which of the following statements is/are correct?

**Options :**

For URL: <http://localhost:5000/?num=abc>

Output is: **Invalid number**

6406531958723.

6406531958724. For URL: <http://localhost:5000/?num=abc>  
Output is: **Enter a valid number**

6406531958725. For URL: <http://localhost:5000/?num=>  
Output is: **Invalid number**

6406531958726. For URL: <http://localhost:5000/?num=12>  
Output is: **The output is 1212**

**Sub-Section Number :** 7  
**Sub-Section Id :** 64065384379  
**Question Shuffling Allowed :** Yes  
**Is Section Default? :** null

**Question Number : 109 Question Id : 640653587014 Question Type : SA Calculator : None**

**Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3**

**Question Label : Short Answer Question**

Consider the following flask application.

```
from flask import Flask, redirect, url_for
app = Flask(__name__)

@app.route('/admin')
def hello_admin():
    return 'Hello Admin'

@app.route('/guest/<guest>')
def hello_guest(guest):
    return 'Hello ' + guest + ' as Guest'

@app.route('/user/<name>')
def hello_user(name):
    if name == 'admin':
        return redirect(url_for('hello_admin'))
    else:
        return redirect(url_for('hello_guest', guest = name))

if __name__ == '__main__':
    app.run(debug = True)
```

If this flask app is running locally on <http://localhost:5000>, what is the output for the following URL?

For input: <http://localhost:5000/user/admin?guest=appdev1>

**Response Type :** Alphanumeric

**Evaluation Required For SA :** Yes

**Show Word Count :** Yes **Answers**

**Type :** Equal **Answers Case**

**Sensitive :** No **Text Areas :**

PlainText **Possible Answers :**

**MLF**

**Section Id :**

64065339714

**Section Number :**

8