

AppDev1

Section Id :	64065360925
Section Number :	3
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	15
Number of Questions to be attempted :	15
Section Marks :	50
Display Number Panel :	Yes
Section Negative Marks :	0
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	No
Section Maximum Duration :	0
Section Minimum Duration :	0
Section Time In :	Minutes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653126833
Question Shuffling Allowed :	No

Question Number : 33 Question Id : 640653852357 Question Type : MCQ

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 :

6406532866951. ✓ YES

6406532866952. ✗ NO

Sub-Section Number :	2
Sub-Section Id :	640653126834
Question Shuffling Allowed :	Yes

Question Number : 34 Question Id : 640653852358 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

Read the statements given below carefully and select the correct option.

Statement 1: The GET method is the same as the HEAD method. It is used to transfer the header section only.

Statement 2: The GET method retrieves information from the given server using a given URI. GET request can retrieve the data.

Options :

6406532866953. ✖ Both statements 1 and 2 are correct

6406532866954. ✖ Both statements 1 and 2 are incorrect

6406532866955. ✖ Statement 1 is correct but statement 2 is incorrect

6406532866956. ✔ Statement 1 is incorrect but statement 2 is correct

Question Number : 35 Question Id : 640653852366 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

Consider the below flask_sqlalchemy data models "Product", "Supplier" and "Supply" given in the code below.

```
class Product(db.Model):
    __tablename__ = "product"
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String, nullable=False)

class Supplier(db.Model):
    __tablename__ = "supplier"
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String, nullable=False)
    address = db.Column(db.String, nullable=True)

class Supply(db.Model):
    __tablename__ = "supply"
    id = db.Column(db.Integer, primary_key=True)
    product_id = db.Column(db.Integer, db.ForeignKey("product.id"))
    supplier_id = db.Column(db.Integer, db.ForeignKey("supplier.id"))
```

What will be the cardinality of the "Product" and "Supplier" relationship?

Options :

6406532866981. ✖ One to One

6406532866982. ✖ One to Many

6406532866983. ✔ Many to Many

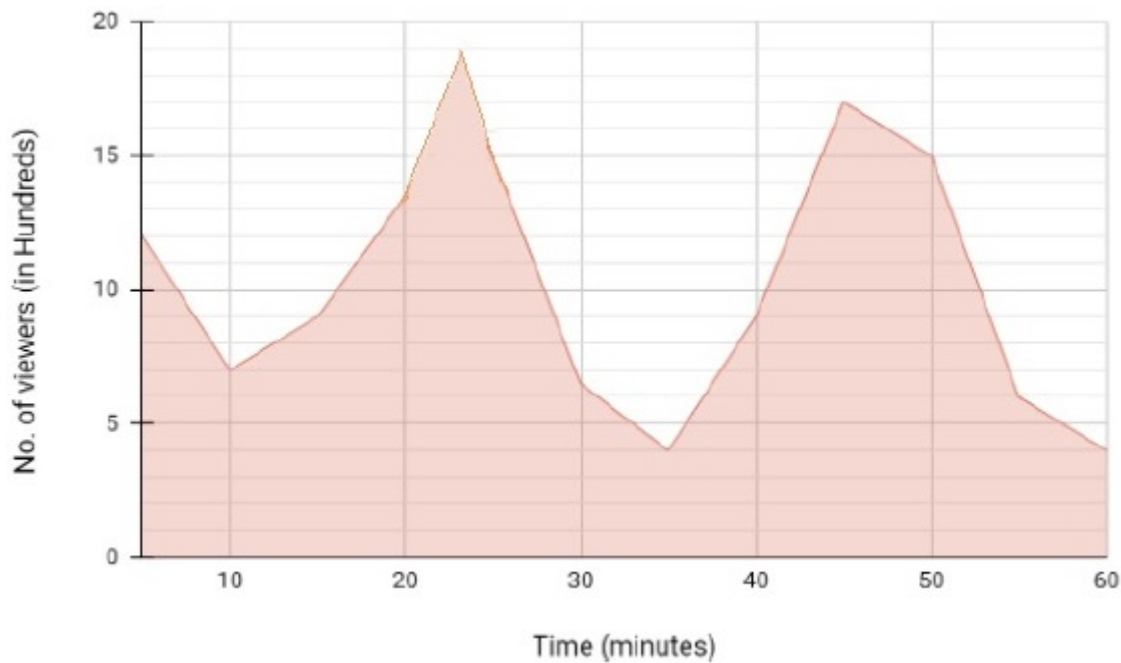
6406532866984. ✖ Many to One

Sub-Section Number :	3
Sub-Section Id :	640653126835
Question Shuffling Allowed :	Yes

Question Number : 36 Question Id : 640653852359 Question Type : MCQ
Correct Marks : 3

Question Label : Multiple Choice Question

A certain video on the web occupies 3 gigabytes of memory of the server. If the number of viewers of the video over time is given by the curve shown below and assuming that each viewer requires an individual connection to the server to view the video, what should be the minimum RAM requirement of the server that can process requests from all the viewers simultaneously at any point of time?



Options :

- 6406532866957. ✖ 1.9 GB
- 6406532866958. ✖ 5.7 GB
- 6406532866959. ✖ 1.9 TB
- 6406532866960. ✔ 5.7 TB

Question Number : 37 Question Id : 640653852372 Question Type : MCQ
Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following Python code.

app.py

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
import os

app_dir = os.path.dirname(os.path.abspath(__file__))
db_file = "sqlite:///course_database.sqlite3"

app = Flask(__name__)
app.app_context().push()
app.config["SQLALCHEMY_DATABASE_URI"] = db_file
app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False

db = SQLAlchemy(app)

class MaxScores(db.Model):
    __tablename__ = "max_scores"
    course_id = db.Column(db.String(10), nullable=False, primary_key=True)
    high_score = db.Column(db.Integer)
    term_year = db.Column(db.String(10))
```

Using *MaxScores* data model in the above “*app.py*”. What is the correct sequence of python commands to insert *MaxScores* record?

Options :

6406532867001. ✓

```
>>> from app.py import *
>>> db.create_all()
>>> s1 = MaxScores(course_id='cs2002', high_score=86, term_year='Jan-2023')
>>> db.session.add(s1)
>>> db.session.commit()
```

6406532867002. ✖

```
>>> from app.py import *
>>> db.create_all()
>>> s1 = MaxScores('cs2002', 86, 'Jan-2023')
>>> db.session.add(s1)
>>> db.session.commit()
```

6406532867003. ✖

```
>>> from app.py import *
>>> db.create_all()
>>> s1 = MaxScores('cs2002', 86, 'Jan-2023')
>>> db.commit()
>>> db.add(s1)
```

```
>>> from app.py import *
>>> db.create_all()
>>> s1 = MaxScores(course_id='cs2002',high_score=86,term_year='Jan-2023')
>>> db.session.commit()
>>> db.session.add(s1)
```

6406532867004. ✖

Question Number : 38 Question Id : 640653852373 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following Python code snippet, and select the correct output list if the code is run on the terminal.

```
def modify(func):
    def wrapper(l):
        res = func(l)
        return res[::-1]
    return wrapper

def update(func):
    def wrapper(l):
        res = func(l)
        return list(map(lambda x: x/2,res))
    return wrapper

@modify
@update
def mylist(l):
    out_list = []
    for i in l:
        out_list.append(i**2)
    return out_list

print(mylist([2,4,1,7,5,9]))
```

Options :

6406532867005. ✖ [0.25, 1, 4, 6.26, 12.25, 20.25]

6406532867006. ✖ [1, 4, 0.25, 12.25, 6.25, 20.25]

6406532867007. ✔ [40.5, 12.5, 24.5, 0.5, 8.0, 2.0]

6406532867008. ✖ [0.5, 2.0, 8.0, 12.5, 24.5, 40.5]

Sub-Section Number :

4

Sub-Section Id :

640653126836

Question Shuffling Allowed :

Yes

Question Number : 39 Question Id : 640653852360 Question Type : MSQ

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following flask application.

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

@app.route('/home')
def home():
    var_a = request.args.get('method')
    if var_a == "GET":
        return """
            <body style="background-color: lightgray; color:
purple;text-align: center;font-weight: bold;">
            <p>IIT</p>
            <p>MADRAS</p>
            </body>
            """
    elif var_a == "POST":
        return """
            <body style="background-color: lightgray; color:
purple;text-align: center;font-weight: bold;">
            <p>IIT</p>
            <span>MADRAS</span>
            </body>
            """
    else:
        return "Invalid Method"
app.run(debug=True)
```

If the application is running locally on `http://127.0.0.1:5000` then which of the following statements are correct?

Options :

6406532866961. ✓ The browser will render aesthetically similar output for `http://127.0.0.1:5000/home?method=POST` and `http://127.0.0.1:5000/home?method=GET`

6406532866962. ✗ The browser will render aesthetically different output for `http://127.0.0.1:5000/home?method=POST` and `http://127.0.0.1:5000/home?method=GET`

6406532866963. ✗ The browser will render `Method Not Allowed` error for `http://127.0.0.1:5000/home`

6406532866964. ✓

The browser will render **Invalid Method** error for <http://127.0.0.1:5000/home>

Question Number : 40 Question Id : 640653852371 Question Type : MSQ

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following code.

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

trains_list = [
    {"train_no": 12842, "source": "Madras", "destination": "Howrah"},
    {"train_no": 12727, "source": "VSKP", "destination": "HYD"},
    {"train_no": 12434, "source": "Delhi", "destination": "Madras"},
]

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

class TrainsApi1(Resource):
    def get(self, tno):
        for train in trains_list:
            if train["train_no"] == tno:
                return train
        return None

class TrainsApi2(Resource):
    def get(self):
        return trains_list

api.add_resource(TrainsApi1, "/api/ver1/<int:tno>")
api.add_resource(TrainsApi2, "/api/ver2/")

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

Assume that the above flask code is running locally on <http://127.0.0.1:5000/>. Which of the below URL(s) requests renders train data without errors?

Options :

6406532866997. ✓ <http://127.0.0.1:5000/api/ver1/12727>

6406532866998. ✗ <http://127.0.0.1:5000/api/12727>

6406532866999. ✗ <http://127.0.0.1:5000/ver2/12727>

6406532867000. ✓ <http://127.0.0.1:5000/api/ver2>

Sub-Section Number :

5

Sub-Section Id :

640653126837

Question Shuffling Allowed :

No

Question Id : 640653852361 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix Question Numbers : (41 to 42)

Question Label : Comprehension

Consider the following flask application running locally on

`http://127.0.0.1:5000`

app.py

```
from flask import Flask, request
import sys
app = Flask(__name__)
data = ["Java", "Application Development", "DBMS"]

@app.route('/course')
def home():
    course = request.args.get('course')
    if course in sys.argv:
        if sys.argv[1] in data:
            return f"Welcome to {sys.argv[1]}!"

        return f"Welcome to {course}!"
    else:
        return "Invalid Data"

app.run(debug=True)
```

Based on the above data, answer the given subquestions.

Sub questions

Question Number : 41 Question Id : 640653852362 Question Type : MSQ

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

What should be the code to run the application, and the URL respectively such that the browser gives output as: `Welcome to Application Development!` ?

Options :

6406532866965. ✖

Code: `python app.py Application Development Java`

URL: `http://127.0.0.1:5000/course?course=Java`

Code: `python app.py Application Development Java`

6406532866966. ✖ **URL:** `http://127.0.0.1:5000/course=Application Development`

Code: `python app.py "Application Development" Java`

6406532866967. ✔ **URL:** `http://127.0.0.1:5000/course?course=Java`

Code: `python app.py "Application Development" DBMS`

6406532866968. ✔ **URL:** `http://127.0.0.1:5000/course?course=Application Development`

Question Number : 42 Question Id : 640653852363 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

What will be the output given by

browser if the application is run with

command `python app.py Application Development DBMS`

on terminal with URL `http://127.0.0.1:5000?course=DBMS` ?

Options :

6406532866969. ✖ `Welcome to Application Development!`

6406532866970. ✖ `Welcome to DBMS!`

6406532866971. ✖ `Invalid Data`

6406532866972. ✔ `Not Found`

Sub-Section Number :

6

Sub-Section Id :

640653126838

Question Shuffling Allowed :

No

Question Id : 640653852368 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix Question Numbers : (43 to 44)

Question Label : Comprehension

If the flask application is running locally on URL `http://127.0.0.1:5000`. Select the appropriate options for the given subquestions.

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

users = [
    {"id": "101", "name": "Ravi", "city": "Chennai"},
    {"id": "102", "name": "Ram", "city": "Mumbai"},
    {"id": "103", "name": "Rahim", "city": "Bhopal"},
    {"id": "104", "name": "Robert", "city": "Delhi"},
]

def search_user(id):
    for user in users:
        if id == user.get("id"):
            return user
    return None

@app.route("/user/<string:id>")
def user(id):
    this_user = search_user(id)
    return this_user

@app.route("/user")
def get_user():
    this_user = search_user(request.args.get('id'))
    return this_user

app.run(debug=True)
```

Sub questions

Question Number : 43 Question Id : 640653852369 Question Type : MSQ Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Which of the following is/are the correct URLs for retrieving the user from the database with `"id": "103"`?

Options :

6406532866989. ✖ <http://127.0.0.1:5000/103>

6406532866990. ✓ <http://127.0.0.1:5000/user/103>

6406532866991. ✗ <http://127.0.0.1:5000?id=103>

6406532866992. ✓ <http://127.0.0.1:5000/user?id=103>

Question Number : 44 Question Id : 640653852370 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

Which of the following URLs will trigger the "get_user" controller in the given application.

Options :

6406532866993. ✗ <http://127.0.0.1:5000/104>

6406532866994. ✗ <http://127.0.0.1:5000/user/104>

6406532866995. ✗ <http://127.0.0.1:5000?id=104>

6406532866996. ✓ <http://127.0.0.1:5000/user?id=104>

Sub-Section Number : 7

Sub-Section Id : 640653126839

Question Shuffling Allowed : Yes

Question Number : 45 Question Id : 640653852364 Question Type : MCQ

Correct Marks : 4.5

Question Label : Multiple Choice Question

You have a DRAM module with bus width of 64 bits, clock speed of 400 MHz, and operating in DDR (double-data-rate or two values per clock cycle) mode. What is the maximum memory bandwidth in Gigabytes per second achievable with this module if it is working in dual channel mode (two memory interfaces are used)?

Options :

6406532866973. ✗ 51.2

6406532866974. ✗ 102.4

6406532866975. ✗ 25.6

6406532866976. ✓ 12.8

Question Number : 46 Question Id : 640653852367 Question Type : MCQ

Correct Marks : 4.5

Question Label : Multiple Choice Question

Two data models “**Student**” and “**Sport**” are to be created in the database with one-to-many relationship using *flask_sqlalchemy*. Which of the following is the correct implementation of the above two data models?

Options :

```
class Student(db.model):
    id=db.Column(db.Integer, primary_key=True)
    Name = db.Column(db.String, nullable=False)
    stud_id=db.Column(db.Integer, db.ForeignKey('sport.id'))
    sports=db.relationship('Sport', backref='student')

class Sport(db.model):
    id=db.Column(db.Integer, primary_key=True)
    sport_name=db.Column(db.String, nullable=False)
```

6406532866985. ✖

```
class Student(db.model):
    id=db.Column(db.Integer, primary_key=True)
    Name = db.Column(db.String, nullable=False)
    sports=db.relationship('Sport', backref='sport')

class Sport(db.model):
    id=db.Column(db.Integer, primary_key=True)
    sport_name=db.Column(db.String, nullable=False)
    stud_id=db.Column(db.Integer, db.ForeignKey('sport.id'))
```

6406532866986. ✖

```
class Student(db.model):
    id=db.Column(db.Integer, primary_key=True)
    Name = db.Column(db.String, nullable=False)
    sports=db.relationship('Sport', backref='student')

class Sport(db.model):
    id=db.Column(db.Integer, foreign_key=True)
    sport_name=db.Column(db.String, nullable=False)
    stud_id=db.Column(db.Integer, db.ForeignKey('student.id'))
```

6406532866987. ✖

6406532866988. ✔

```

class Student(db.model):
    id=db.Column(db.Integer, primary_key=True)
    Name = db.Column(db.String, nullable=False)
    sports=db.relationship('Sport', backref='student')

class Sport(db.model):
    id=db.Column(db.Integer, primary_key=True)
    sport_name=db.Column(db.String, nullable=False)
    stud_id=db.Column(db.Integer, db.ForeignKey('student.id'))

```

Question Number : 47 Question Id : 640653852375 Question Type : MCQ

Correct Marks : 4.5

Question Label : Multiple Choice Question

Consider the following Python code snippet.

Filename: code.py

```

import logging
import sys

logging.basicConfig(format='%(levelname)s:%(message)s', level=logging.INFO)

logging.warning('This is only for testing.')

if sys.argv[1] == 'info':
    logging.info('This code is working as expected.')

if sys.argv[1] == 'debug':
    logging.debug('This code may have issues, debugging...')

logging.warning('code execution complete!')

```

What will be output on the terminal if the above code snippet is run on the terminal using command: `python code.py debug`

Options :

```

WARNING:This is only for testing.
INFO: This code is working as expected
WARNING:code execution complete!

```

6406532867013. ✖

```

WARNING:This is only for testing.
DEBUG: This code may have issues, debugging...
WARNING:code execution complete!

```

6406532867014. ✖

6406532867015. ✓

```
WARNING:This is only for testing.  
WARNING:code execution complete!
```

6406532867016. ✖

```
INFO: This code is working as expected  
DEBUG: This code may have issues, debugging...
```

Sub-Section Number : 8
Sub-Section Id : 640653126840
Question Shuffling Allowed : Yes

Question Number : 48 Question Id : 640653852365 Question Type : MSQ

Correct Marks : 2 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following flask application.

app.py

```
from flask import Flask, abort, request  
app = Flask(__name__)  
  
@app.route('/login/<id>')  
def login(id):  
    if id:  
        if id.isalpha():  
            abort(400, "Bad Request: Invalid ID")  
        return f'<h1>Your ID is: {id}</h1>'  
    return f'<h1>Invalid ID</h1>'  
  
app.run(debug=True)
```

Which of the following statements is/are true if the application is running locally on <http://127.0.0.1:5000> ?

Options :

For URL `http://127.0.0.1:5000/login?id=MAD-I` the browser will render
Bad Request: Invalid ID

6406532866977. ✖

For URL `http://127.0.0.1:5000/login/` the browser will render
Invalid ID

6406532866978. ✖

6406532866979. ✓

For URL `http://127.0.0.1:5000/login/CS2003` the browser will render

Your ID is: CS2003

For URL `http://127.0.0.1:5000/login/PDSA` the browser will render Bad

Request: Invalid ID

6406532866980. ✓

Sub-Section Number : 9
Sub-Section Id : 640653126841
Question Shuffling Allowed : Yes

Question Number : 49 Question Id : 640653852374 Question Type : MSQ

Correct Marks : 4.5 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following HTML Document and select the correct statement(s) from the following if a user types the password "mad1password" in the password field letter by letter.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Login Form</title>
  <style>
    input[type="password"]:valid{
      background-color: green;
    }
    input[type="password"]:invalid{
      background-color: red;
    }
  </style>
</head>
<body>
  <div class="login-form">
    <h2>Login</h2>
    <form>
      E-mail: <input type="email" id="email" name="email" required>
      Password: <input type="password" id="password" name="password"
        required minlength="5" maxlength="8">
      <input type="submit" value="Login">
    </form>
  </div>
</body>
</html>
```

Options :

6406532867009. ✗ The HTML form is integrated with backend validation.

6406532867010. ✓ The HTML form is integrated with Frontend HTML5 validation.

6406532867011. ✖ The background colour of the password field remains red for the first 4 letters, then turns green for the next 4 letters, and then again turns red for the remaining letters.

6406532867012. ✔ The background colour of the password field remains red for the first 4 letters, then turns green for the next 4 letters, and then remains green for the remaining allowable letters.

MLF

Section Id :	64065360926
Section Number :	4
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	11
Number of Questions to be attempted :	11
Section Marks :	40
Display Number Panel :	Yes
Section Negative Marks :	0
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	No
Section Maximum Duration :	0
Section Minimum Duration :	0
Section Time In :	Minutes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653126842
Question Shuffling Allowed :	No

Question Number : 50 Question Id : 640653852376 Question Type : MCQ

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : MACHINE LEARNING FOUNDATIONS (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 :

6406532867017. ✔ YES

6406532867018. ✖ NO