

INTEGRITY AND RESPECT

Devang Gandhi

ACADEMIC INTEGRITY



**KNOW THE CODE
It Is YOUR Responsibility**

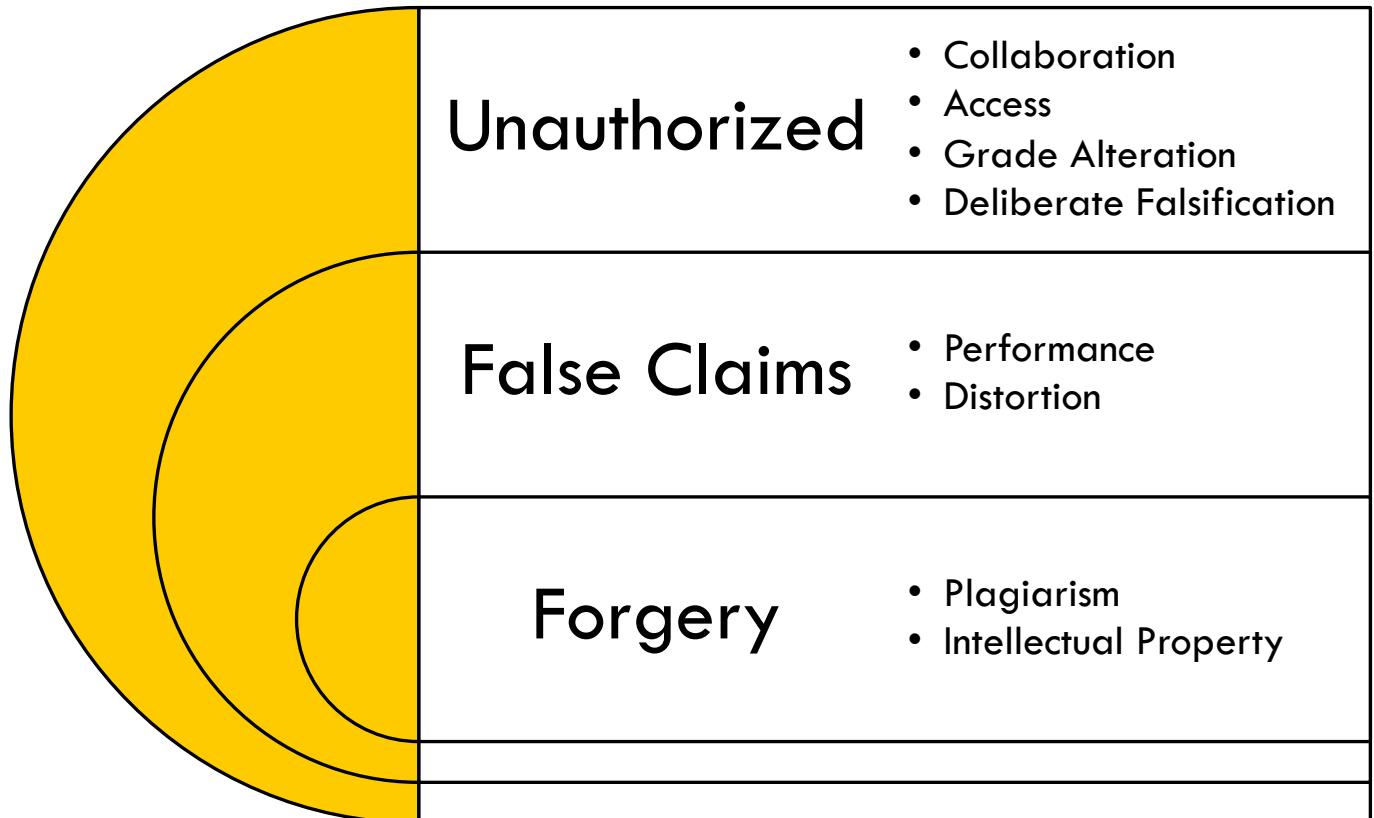
<http://www.policylibrary.gatech.edu/student-life/student-conduct>

(Source: <http://osi.gatech.edu/>)

ACADEMIC MISCONDUCT



(Source: Brigham Young University)



- Collaboration
- Access
- Grade Alteration
- Deliberate Falsification

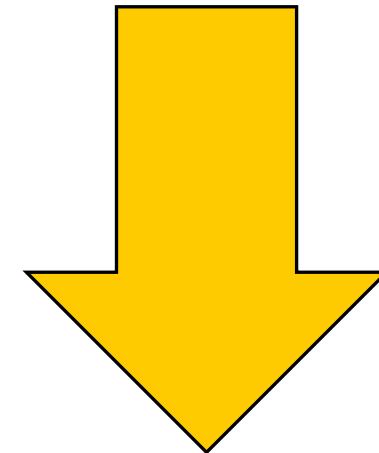
- Performance
- Distortion

- Plagiarism
- Intellectual Property

<http://www.policylibrary.gatech.edu/student-life/academic-misconduct>

WHO CARES?

Issue of
fairness



Inflated
perception
of abilities

TABLE 5. Correlations Between Dishonesty at School and at Work for Graduate and Undergraduate Students

	Pearson's correlation	Partial correlation ^a
Independent variable: Frequency of academic dishonesty		
Undergraduates	0.66	0.62
Graduates	0.61	0.60

^aPartial correlation after controlling for the demographic variables gender and age.

“... Pearson's ... correlation [coefficient] ... support, that the ... academic dishonesty and work dis-honesty are positively correlated ($p < .005$)”

-Nonis and Owens Swift (2001)

BUT YOU SAID, “GOOGLE IT!”

(Source: Brigham Young University)



...I also said to use it to build your pseudocode chops.

Software industry (Data science is part of it) thrives on strong and supportive user community to share and learn from each other. Therefore it is of utmost important that we remain vigilant about how we use the information we obtain while learning.

It must be your thoughts, your logic, your code!

Your work

Pseudocode
... your code

Learning from
others

Boiler Plate
Code

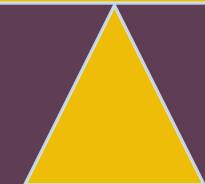
Plagiarism

It was only a
bit I kept

“the var’s are
different”

Copy...Paste

It's a fine balance between the two!



BOILER PLATE CODE

```
from sqlalchemy.ext.automap import automap_base  
from sqlalchemy.orm import Session  
from sqlalchemy import create_engine  
Base = automap_base()  
create_engine("sqlite:///mydatabase.db")  
Base.prepare(engine, reflect=True)
```

(<https://docs.sqlalchemy.org/en/latest/orm/extensions/automap.html>)

It's a specific piece of code as given by the publisher of a library or a software, that must be used in the exact manner to enable access to the tool or a package and can't be written any other way

COPY...PASTE

```
User = Base.classes.user  
  
Address = Base.classes.address  
  
session = Session(engine)  
  
session.add(Address(email_address="foo@bar.com", user=User(name="foo")))  
  
session.commit()  
  
print (u1.address_collection)
```

(<https://docs.sqlalchemy.org/en/latest/orm/extensions/automap.html>)

Code that solves a very specific problem and typically isn't useful verbatim, unless applied to the very problem. Usually not published or available from a legal source such as Stack Overflow

CAN I SIMPLY CHANGE SOME VARIABLES?

```
My_User = Base.classes.user  
  
My_Address = Base.classes.address  
  
My_session = Session(engine)  
  
My_session.add(My_Address(email_address="foo@bar.com", user=My_User(name="foo")))  
  
My_session.commit()  
  
print (u1.address_collection)
```

(<https://docs.sqlalchemy.org/en/latest/orm/extensions/automap.html>)

Not your thoughts, not your logic, **NOT ENOUGH!**

SO WHAT ARE MY OPTIONS?

```
User = Base.classes.user  
  
Address = Base.classes.address  
  
session = Session(engine)  
  
session.add(Address(email_address="foo@bar.com", user=User(name="foo")))  
  
session.commit()  
  
print (u1.address_collection)
```

(<https://docs.sqlalchemy.org/en/latest/orm/extensions/automap.html>)

Be sure the source of your code is intended for sharing and is not published for nefarious reasons.

LEARN!

```
% use classes field in the Base variable to obtain my table class for each table  
% (User and Address), example User = Base.classes.user  
  
% create a session from the engine  
  
% create instances of my table classes with corresponding data example  
% user=User(name="foo")  
  
% add it to session using session.add(user) and commit it using session.commit()  
% not sure what this does. Need to investigate...print (u1.address_collection)
```

Pseudocode it into your own understanding

WRITE YOUR OWN!

```
% create a session from the engine
session = Session(engine)
% use classes field in the Base variable to obtain my table class for each
table (User and Address)
User = Base.classes.user
Address = Base.classes.address
% create instances of my table classes with corresponding data
user1 = User(name="dev")
address_u1 = Address(email_address=dev@abc.com, user=user1)
% add it to session using session.add() and commit it using session.commit()
session.add(user1)
session.add(address_u1)
session.commit()
% not sure what this does. I'll keep it...
print(u1.address_collection)
```



RESPECT

1. Respect is not reserved for age, status or rank.
2. Respect exist from the onset of existence until lost by misdeeds. Therefore, 65-year-old professor if has cheated and lied must deserve less respect than a “23-year-old boys scout.” But do not disrespect the boys scout for the age.
3. It is a right of every student to learn and ask questions, and by entering a classroom we are honor bound to respect that
4. Your lack of understanding is not by because someone asked a question, but because you didn’t.

Listen, Ask and Learn!