DATA 533: Collaborative Software Development

Instructions:

1. Write your answer in the space provided.
2. You may use class lectures on your computer. You may use the GitHub repository for the course.
3. **You are NOT allowed to use Jupyter Notebook or any python program to compile your program. You are not allowed to use any other online resources.**
4. In the quiz, you will have 60 minutes to complete

Part 1: Short Question (Note: there will be 15-20 short questions on Quiz 1)

1. What is the output of the following program?

```
class C1(object):
  def __init__(self):
    self.i = 1

class C2(C1):
  def __init__(self):
    C1.__init__(self)
    self.j = 2

a = C1()
b = C2()

print(a.i, b.i, b.j)
```

Output:1 1 2

2. What output would be produced by the following program:

```
class Widget:
    def __init__(self, width, height):
        self.width = width
        self.height = height
```

```
    def process(self, width, height):
        print("Width:", self.width, "Height:", self.height)

w = Widget(10, 20)
w.process(15)
```

Output: Error due to the missing argument: 'height'

3. Write the output of the following program. If the code encounters an error/exception, write the type of error/exception.

```
try:
    print(float('data533'))
except NameError:
    print("NameError")
except TypeError:
    print("TypeError")
except ValueError:
    print("ValueError")
else:
    print ('Success, no error!')
```

Output: ValueError

4. Write the output of the following program. If the code encounters an error/exception, write the type of error/exception.

```
def myFunc(x):
    try:
        y = x / x
    except:
        print("1")
    else:
        print("2")
    finally:
        print("3")
myFunc(1)
myFunc(0)
```

Output: 2 3 1 3

5. Write the output of the following program. If the code encounters an error/exception, write

the type of error/exception.

```
def addition():
    try:
        num = 20
        results += num
        return results
    except NameError:
        return "NameError"
    except TypeError:
        return "TypeError"
    except ValueError:
        return "ValueError"
    except :
        return "Exception occured"

print(addition())
```

Output: NameError

6. Write the output of the following program. If the code encounters an error/exception, write the type of error/exception.

```
import math

try:
    print(math.exp(1000))
except Exception:
    print("Exception")
except ArithmeticError:
    print("ArithmeticError")
except OverflowError:
    print("OverflowError")
except :
    print("Exception occured")
else:
    print("Success")
```

Output: Exception

7. Consider the following scenario involving Git and GitHub. Alex and William are collaborating on a project. They stored their project in a remote Git repository (e.g., GitHub). Alex wants to create a copy of the repository and save it in the local workspace. What command does he need to issue?

git clone

Part 2: Multiple-Choice Question (Note there will be 10 multiple-choice questions)

1. What will be the output of the following program?

```
class Person:
    def __init__(self, name):
        self.name = name
        name = 'William'

val = Person('Alex')
print(val.name)
```

a. Error, this program will not run

b. Alex

c. William

d. None of the above

2. What will be the output of the following program?

```
class A:
    def __init
        (self):
        self.i = 10


class B(A):
    def __init
        (self): A.
        init__(self)
        self.i = 20
        print(self.i)

b = B()
```

a. Error, this program will not run

b. 10

c. 20

d. None of the above

3. Let's assume that Ana and Alex are collaborating on a project and both copied a remote repository to their local repository. Ana has pushed some changes into a remote repository. What Git command would Alex use to download the changes into his local repository?

a. clone

b. fork

c. pull

d. branch

e. None of the above

4. Which is the output of the following program?

```
class MyLanguage:
  def language(self,
    lang='Python'): print(lang)

lang1 = MyLanguage()
lang1.language('Java')
```

a. Python

b. lang

c. Java

d. Java Python

5. Which of the following statement(s) is/are TRUE? Please write the correct answer(s) (e.g., A, B, C)

A. Black box testing examines the program source code in order to identify potentially problematic sets of inputs

B. White box testing validates programs without any knowledge of how the system is implemented

C. Integration tests exercise groups of components to ensure that their contained units interact correctly together.

D. UI testing can be done with a human tester to verify that a program is behaving correctly

a. A, B

b. A, C

c. C, D

d. B, C

e. None of the above

6. Which of the following statement(s) is/are TRUE? Please write the correct answer(s) (e.g., A, B, C)

A. In unittest, to make our own test cases, we need to write subclasses of *TestCase*
B. A test case can be a collection of test suites
C. We can run tests with more detailed information by passing in the verbosity argument
D. In unittest, if the setUp() method raises an exception, test methods will not be executed

a. A, B

b. A, C

c. C, D

d. B, C

e. None of the above

Part 3: Coding (Note there will be 2 coding questions)

The following code creates a class called Rocket that takes self, name, and manufacturer as arguments in its init () method. It also has a method called display in the class to show the Rocket's member variables (sample format: [Name] is manufactured [manufacturer]). Now complete the missing lines (1), (2) and 3 so that the program shows the output for the following test code:

Test code:

```
x = Falcon("Falcon 9 CRS-16", "SpaceX", "2018")
 x.display()
```

Output:

Falcon 9 CRS-16 is manufactured by SpaceX Falcon 9 CRS-16 launched date 2018

```
class Rocket:
        def __init__ (self, name, manufacturer):
                self.name = name
                self.manufacturer = manufacturer

def display(self):
        print("%s is manufactured by %s" % (self.name, self.manufacturer))

class Falcon(Rocket):
```

```
def __init__ (self, name, manufacturer, date):
        (1) _____
        (2) _____

def display(self):
        Rocket.display(self)
        (3) _____
```

(1) Rocket. init (self, name, manufacturer)
(2) self.date = date
(3) print("%s launched date %s" % (self.name, self.date))