

Name:

Python Code

```
def combine(a, b):
    return a + b

R2D2 = 'droid'
C3P0 = 'droid'
Leia = 'Princess'
DarthVader = 'Best Villain Ever'

print "The " + Leia + " entrusted a " + R2D2 + " with the secret plans "

Leia = combine( 'Captured ', 'by the Empire')
DarthVader = combine( 'The Force is ', 'strong')
R2D2 = combine( 'Property of ', ' Obi-wan Kenobi')
C3P0 = combine( "Chances of success are 1421 to ", "1" )

print 'The droids landed on Tatooine ' + str(100) + ' miles from civilization.'
```

Questions

1. (8 points) Write down a representation of the global symbol table (name, type, value) at the last line of code. The symbol table should include all identifiers defined by the code up to that point.
2. (4 points) What does the program print out when it is executed?
3. (4 points) In what symbol table do the parameters `a` and `b` belong?
4. (4 points) Would the following expression work? Why/why not?
`combine("Chances of success are 1421 to ", 1)`

Name:

Python Code

```
def jawa( droid1, droid2 ):
    return droid1 + ' ' + droid2

def UncleOwen( ):
    print 'Stay another cycle.'

R2D2 = "heads west"
C3PO = "heads east"
sandcrawler = jawa( 'R2D2', 'C3PO' )
for i in range(100):
    sandcrawler = jawa( sandcrawler, "and another droid" )

Owen = jawa( 'R2D2', 'C3PO' )
Luke = UncleOwen()

print 'I am the property of Obi-wan Kenobi' # mark 1

if R2D2 == "heads west":
    R2D2 = R2D2 + " and finds Ben Kenobi"

C3PO = "broken"
Luke = "concussed" # mark 2
```

Questions

1. (4 points) What is the value of the variable Luke at # mark 1?
2. (8 points) List all of the symbols in the global symbol table at the end of the code (no types/values).
3. (4 points) Describe the value of the symbol sandcrawler at # mark 2. Do not write it all out.
4. (4 points) What is the value of the symbol R2D2 at the end of the code?

Name:

Python Code

```
companions = ['Luke', 'R2D2', 'C3PO', 'Ben']
homestead = ['Owen', 'Beru']
cantina = [ 'Han Solo', 'Chewbacca', 'Greedo' ]
jawas = []
for i in range(100):
    jawas.append( 'jawa' + str(i) )
deathstar = ['Darth Vader', 'Leia', 'Tarkin', 'Commander']
for i in range(50000):
    deathstar.append( 'Stormtrooper' )

print companions[-1] + ' explains the force to ' + companions[0]

for i in range(len(jawas)):
    jawas[i] = 'ex-' + jawas[i]

homestead[0] = 'burned'
homestead[1] = 'burned'
print companions[-4] + ' returns home and finds it ' + homestead[1]

cantina = cantina[0:2]
companions.append(cantina[0])
companions.append(cantina[1]) # mark 1
print companions[-1] + ' and ' + companions[-2] + ' agree to help ' + companions[0]
cantina = []

print 'Meanwhile, ' + deathstar[0] + ' does not like the ' + deathstar[3]
deathstar[3] = 'almost ex-' + deathstar[-50001]

print deathstar[2] + ' comes up with a plan to break ' + deathstar[-50000 - 3]
```

Questions

1. (3 points) How many elements does the list `deathstar` contain at the end of the code?
2. (3 points) What does the list `cantina` contain at # mark 1?
3. (4 points) How many elements does the list `jawas` contain at the end of the code?
4. (10 points) What does the program print? There are five print statements.

Name:

Python Code

```
def bzzzzSwoosh( knights ):
    return knights.pop()

def zapFlashBang( thugs ):
    thugs.pop()

def ABiggerChallenge( milleniumFalcon ):
    force = ['Qui-gon Jinn']
    deathstar = [ 'Docking Bay', 'Darth Vader', 'Leia', 'Tarkin' ]
    for i in range(50000):
        deathstar.append( 'Stormtrooper' )

    milleniumFalcon[0] = milleniumFalcon[0] + ", Jedi Apprentice"
    milleniumFalcon[-1] = milleniumFalcon[-1] + ", Jedi Master"
    deathstar[0] = milleniumFalcon
    print deathstar[0][3] + ' discovers ' + deathstar[2]

    for extras in range(10): # mark 1
        zapFlashBang( deathstar )
    print deathstar[0][2]+' and '+deathstar[0][1]+' help '+deathstar[0][0][0:4]

    print deathstar[1] + ' confronts ' + deathstar[0][-1]
    force.append( bzzzzSwoosh( deathstar[0] ) )

    for more_extras in range(10):
        zapFlashBang( deathstar )
    milleniumFalcon.append( deathstar.pop( 2 ) )
    deathstar[0] = 'Docking Bay'
    print milleniumFalcon[-1] + ' is not impressed.' # mark 2

if __name__ == "__main__":
    characters = [ 'Luke', 'Han Solo', 'Chewbacca', 'R2D2', 'C3PO', 'Ben Kenobi' ]
    ABiggerChallenge( characters )
```

Questions

1. (4 points) How does the for loop at mark 1 modify deathstar?
2. (4 points) What is the value of mileniumFalcon at mark 2?
3. (4 points) What is the value of force at mark 2?
4. (8 points) What does the function ABiggerChallenge print, given the code (4 statements)?

Name:

Python Code

```
def PlanAttack( rebels ):  
    secret = rebels[-1][-1][5:]  
    rebels[-1][-1] = rebels[-1][-1][0:4]  
    return 1.0 / 1000.0  
  
def PrepareForAssault( pilots ):  
    return [pilots.pop(), pilots.pop()]  
  
def TheRebelBaseWillBeInRange( milleniumFalcon ):  
    rebelBase = [ ['General Willard'], ['Wedge', 'droid'], ['Biggs', 'droid'] ]  
    rebelBase.append( milleniumFalcon.pop() )  
    rebelBase.append( milleniumFalcon.pop() )  
  
    chances = PlanAttack( rebelBase )  
    if chances < 1.0 / 10.0:  
        print milleniumFalcon[0][0] + " tells " + rebelBase[3][0] + " it's suicide."  
        milleniumFalcon.append( ['reward', 'more reward'] )  
  
    s = rebelBase[1][0] + ", " + rebelBase[2][0] + ", and "  
    print s + rebelBase[-1][0] + " prepare."  
    redTwo = PrepareForAssault( rebelBase[1] )  
    redThree = PrepareForAssault( rebelBase[2] )  
    redFive = PrepareForAssault( rebelBase[4] )  
    rebelBase = [rebelBase[0], rebelBase[3]]  
  
    print rebelBase[1][0] + " wishes " + redFive[1] + " good luck."  
    redSquadron = [ redTwo, redThree, redFive]  
    print milleniumFalcon[-2][0] + ' says may the force be with you.' # mark 2  
  
players = [ ['Han Solo', 'Chewbacca'], ['Luke', 'R2D2 plans'], ['Leia', 'C3PO'] ]  
TheRebelBaseWillBeInRange( players )
```

Questions

1. (4 points) Write out the list milleniumFalcon at # mark 2. Use Python syntax.
2. (4 points) Write out the list redSquadron at # mark 2. Use Python syntax.
3. (4 points) Write out the list rebelBase at # mark 2. Use Python syntax.
4. (8 points) What does the program print when executed (4 print statements)?

Name:

Python Code

```
def Vader( rebels ):
    target = rebels[-1]
    if 'L' in target[0]:
        target[1] = target[1] + " out of commission"
    elif 'W' in target[0]:
        target[0] = target[0] + " breaking off"
    return rebels[:-1], rebels[-1]

def surprise( empire ):
    empire.pop()
    empire.pop()
    return empire[0], [ 'Han Solo', 'Chewbacca' ]

def StartYourRun( xwings ):
    TIEfighters = [ ['Vader', 'black TIE'], ['Stormtrooper'], ['Stormtrooper'] ]
    deathstar = [ 'Tarkin', 'Commander' ]
    for i in range(50000):
        deathstar.append( 'Stormtrooper' )

    xwings, lost = Vader( xwings )
    xwings, retreating = Vader( xwings )
    xwings, lastHope = Vader( xwings )
    onlySurvivor, milleniumFalcon = surprise( TIEfighters )

    deathstar = ['debris field'] # mark 1
    print milleniumFalcon[0] + ' tells ' + lastHope[0] + ' it was a good shot.'

rebelShips = [ [ 'Luke', 'R2D2' ], [ 'Wedge', 'droid' ], [ 'Biggs', 'droid' ] ]
StartYourRun( rebelShips )
```

Questions

1. (10 points) List the name and type of all symbols in the `StartYourRun` symbol table at mark 1.
2. (4 points) What is the value of `onlySurvivor` at mark 1?
3. (2 points) What is the value of `i` at mark 1?
4. (4 points) What does the last line of `StartYourRun` print?

Name:

Python Code

```
def lightSaber(thing):
    return thing[:-1]

def blastIt( thing ):
    thing.pop()

def discovery( scouts ):
    cave = [ 'Yeti', scouts[0] ]
    surface = [cave.pop(), scouts[1], 'imperial droid', scouts[2]]
    surface = lightSaber( surface )
    blastIt( surface )
    return surface # mark 1

def battle( rebels, empire ):
    blastIt( empire )
    blastIt( empire )
    empire = lightSaber( empire )
    blastIt( rebels )
    return empire, rebels

def escape( actors ):
    group1 = " "
    for actor in actors[1:3]:
        group1 += actor + ", "
    return group1+"and "+actors[3], actors[0]+" and "+actors[4] # mark 2

def Hoth( empire ):
    rebels = discovery( [ 'Luke', 'Han Solo', 'tuantuan' ] )
    empire, rebels = battle( rebels+[ 'Leia', 'C3PO', 'R2D2', 'Shield Generator'], empire )
    MilleniumFalcon, Xwing = escape( rebels )

    fp = file( 'Apology', 'w' )
    fp.write( 'Dear '+empire[0]+"\\n"+"I'm sorry for letting"+ MilleniumFalcon+' escape.\\n')
    fp.close() # mark 3

Hoth( ['Darth Vader', 'Star destroyer', 'Star destroyer', 'Walker', 'Walker', 'Walker'] )
```

Questions

1. (6 points) Write the name and type of each symbol in the `discovery` symbol table at mark 1.
2. (6 points) Write the name and type of each symbol in the `escape` symbol table at mark 2.
3. (5 points) Write the name and type of each symbol in the `Hoth` symbol table at mark 3.
4. (4 points) What is written to the file `Apology`?

Name:

Python Code

```
class Jedi:
    def __init__(self, name, side = 'Dark', force = 1):
        self.name = name
        self.side = side
        self.force = force

    def deed(self, action, believe, power):
        if believe and self.force >= power:
            return self.name + ' did ' + action
        else:
            return self.name + ' did not ' + action

    def train(self, amount):
        self.force += amount

def Dagobah():
    Luke = Jedi( 'Luke', 'Unknown', 10 )
    Yoda = Jedi( 'Yoda', 'Light', 100000 )
    Ben = Jedi( 'Ben Kenobi', 'Light', 1 )
    VaderSpirit = Jedi( 'Faux Vader', 'Dark', 900 )

    Luke.train( 100 )
    print Luke.deed( 'run through the jungle', True, 50 )
    Luke.train( 1000 )
    print Luke.deed( 'lift and balance heavy rocks', True, 80 )

    Luke.train( 1000 )
    print Luke.deed( 'lift the X-wing', False, 1000 ) # mark 1
    print Yoda.deed( 'lift the X-wing', True, 1000 )
    print Luke.deed( "confront Vader's likeness", True, VaderSpirit.force ) # mark 2
    print Ben.deed( "say he could not help Luke confront Vader", True, 1 )

Dagobah()
```

Questions

1. (6 points) Write out the Jedi class symbol table, name and type only.
2. (6 points) Write out the symbol table—name, type, and value—for the object referenced by the variable Luke at mark 1.
3. (4 points) What does the program print on the line at mark 1?
4. (4 points) Write out the symbol table—name, type, and value—for the deed method just before it returns, as it is called at mark 2.

Name:

Python Code

```
class Place:
    def __init__(self, name, actors, status):
        self.name = name
        self.actors = {}
        for i in range(len(actors)):
            self.actors[actors[i]] = status[i]

    def update(self, actor, newstatus):
        self.actors[actor] = newstatus

    def updateAll(self, newstatus):
        for key in self.actors.keys():
            self.actors[key] = newstatus

    def getStatus(self, actor):
        return actor + ' is ' + self.actors.get(actor, 'not there.')
```

```
def Adventures(MF):
    Asteroid = Place( 'Asteroid', MF, ['grumpy', 'ok', 'ok', 'ok', 'ok'] )
    Asteroid.update( 'Leia', 'smitten' )
    Asteroid.update( 'C3PO', 'concerned' )
    Asteroid.updateAll( 'concerned they will be eaten.' )
    print Asteroid.getStatus('Han Solo')
    print Asteroid.getStatus('Luke') # mark 1

    CloudCity = Place( 'Cloud City', MF, ['suspicious', 'worried', 'ok', 'ok', 'ok'] )
    CloudCity.update( 'C3PO', 'in pieces' )
    CloudCity.update( 'Han Solo', 'captured and tortured' )
    CloudCity.update( 'Han Solo', 'encased in carbonite' )
    print CloudCity.getStatus( 'Han Solo' )
    print CloudCity.getStatus( 'Leia' )
    print CloudCity.getStatus( 'C3PO' ) # mark 2

MilleniumFalcon = ['Han Solo', 'Leia', 'Chewbacca', 'R2D2', 'C3PO']
Adventures(MilleniumFalcon)
```

Questions

1. (5 points) Give the name and type of each symbol in the Place class symbol table.
2. (5 points) Give the name and type of each symbol in the object referenced by Asteroid at mark 1.
3. (5 points) Given the name and type of each symbol in the Adventures function symbol table at mark 2.
4. (5 points) What does the program print?

Name:

Python Code

```
class Place:
    def __init__(self, name, who, status):
        self.noun = name
        self.adjectives = {}
        self.matcher( who, status )

    def matcher(self, properNouns, modifiers):
        for i in range(len(properNouns)):
            self.adjectives[properNouns[i]] = modifiers[i]

    def update(self, fred, george):
        self.adjectives[fred] = george

class City(Place):
    def __init__(self, name, planet, people, state):
        Place.__init__(self, name, people, state)
        self.context = planet

class Ship(Place):
    def __init__(self, name, repair, people, state):
        Place.__init__(self, name, people, state)
        self.works = repair

def escape(actors):
    CloudCity = City( 'Cloud City', 'Bespin', actors, ['ok']*7 ) # mark 1
    CloudCity.update( 'C3PO', 'partially repaired' )
    CloudCity.update( 'Han Solo', 'frozen in carbonite' )
    CloudCity.update( 'Boba Fett', 'pleased' )
    OddShip = Ship( "Boba Fett's Ship", 'working', actors[5:], ['not', 'awake'] )
    CloudCity.update( 'Lando', 'not pleased' )
    MilleniumFalcon = Ship('Millenium Falcon', 'disabled', actors[:5], ['fleeing']*5 )

escape( ['Chewbacca', 'Leia', 'C3PO', 'R2D2', 'Lando', 'Han Solo', 'Boba Fett'] )
```

Questions

1. (5 points) Identify all of the functions called in this code order to execute the line of code at mark 1.
2. (5 points) Write down the value of the expression `OddShip.adjectives` at the end of the `escape` function.
3. (5 points) Write down the name and type of all entries in the object referenced by `MilleniumFalcon` at the end of the `escape` function.
4. (5 points) Write down the name and type of all entries in the symbol table of the `__init__` method of the `City` class when it is called at mark 1.

Name:

Python Code

```
def duel( master, apprentice ):
    if apprentice['health'] <= 1:
        print master['name'], ': I am your father.'
        print apprentice['name'], ': Nooooooo!'
        return ' jumps'
    else:
        apprentice['health'] -= master['skill'] - apprentice['skill']
        print master['name'], ': You have no chance.'
        print apprentice['name'], ": You can't catch me."
        return duel(master, apprentice)

def repair( ship, robot ):
    if len(ship) == 1:
        return robot['name'] + ' turns the right ' + ship.pop()
    else:
        print robot['partner'], 'tests the', ship.pop()
        return repair( ship, robot )

def VadersPlan(MF):
    Vader = { 'name':'Darth Vader', 'health':100, 'skill':55 }
    Luke = { 'name': 'Luke', 'health':151, 'skill': 5 }

    print Luke['name'] + duel( Vader, Luke )

    print Luke['name'], 'calls for', MF[0]['name'] # mark 1
    MF.append( Luke )
    print repair( ['lever', 'coils', 'power supply', 'gasket'], MF[4] ) # mark 2

MilleniumFalcon = [ {'name':'Leia'}, {'name':'Lando'}, {'name': 'Chewbacca'},
                    {'name':'C3PO'}, {'name': 'R2D2', 'partner':'Chewbacca'}]
VadersPlan(MilleniumFalcon)
```

Questions

1. (5 points) Explain why the duel function eventually terminates and how many times it executes.
2. (5 points) What does the line at mark 1 print?
3. (5 points) What does the line at mark 2 print (including statements in repair)?
4. (5 points) Give the name and type of all symbols in the VadersPlan symbol table.

Name:

Python Code

```
def FirstScene(LastArrival):
    JabbasPalace = {'Jabba':'big slug', 'Translator':'smooth talker',
                   'Rancor':'big and ugly', 'Muppet Rat':'strange',
                   'Han Solo':'wall decoration', 'Lando':'disguised'}

    Visitors = {'Leia':'disguised', 'Chewy':'in handcuffs'}

    for visitor in Visitors.keys():
        JabbasPalace[visitor] = Visitors[visitor]

    s1 = 'Leia arrives '+JabbasPalace['Leia']+' with Chewbacca '+JabbasPalace['Chewy']

    JabbasPalace['Chewy'] = 'in a cell'

    MoreVisitors = {'R2D2':'in on the plan', 'C3PO':'clueless'}

    for droid in MoreVisitors.keys():
        JabbasPalace[droid] = MoreVisitors[droid]

    droids = MoreVisitors.keys()
    droids.sort()

    s2 = droids[0]+' and '+droids[1]+' arrive. '+droids[1]+' is '+JabbasPalace[droids[1]]

    JabbasPalace['C3PO'] = 'Protocol droid'
    JabbasPalace['Leia'] = 'undisguised'
    JabbasPalace['Han Solo'] = 'unfrozen but discombobulated'
    JabbasPalace['Jabba'] = 'not happy'
    JabbasPalace['Leia'] = 'in chains'
    JabbasPalace['Han Solo'] = JabbasPalace['Chewy'] + ', and blind'

    JabbasPalace[LastArrival] = "thinks he's a Jedi"
    # mark 1
    return s1 + "\n" + s2

print FirstScene('Luke')
```

Questions

1. (10 points) List the name and type of each entry in the FirstScene function symbol table.
2. (5 points) At mark 1, what is the value of JabbasPalace['Han Solo']?
3. (5 points) What does the whole program print?