

**Name:**

---

### Python Code

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

HarryPotter = 'baby'
JamesPotter = 'dad'
LilyPotter = 'mom'
Valdemort = 6

print "Harry's "+LilyPotter+" and "+JamesPotter+" protected their "+HarryPotter

JamesPotter = combine( 'Avada Kedavra', ' death')
LilyPotter = combine( 'Avada Kadavra', ' death')
HarryPotter = combine( 'Avada Kadavra', ' scar')
Valdemort = combine( Valdemort, 1 )

print 'Harry went to live with his Aunt and Uncle for ' + str(7) + ' years.'
```

---

### 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 expression `combine(6, ' horcruxes')` work? Why/why not?

**Name:**

---

**Python Code**

```
def owl( thing, message ):
    return thing + ' ' + message

def UncleVernon( letter ):
    print 'He rips up the letter'

Dudley = 'pig'
letter = owl( 'Hogwarts', 'Invitation' )
Harry = UncleVernon( letter )
# mark 1

collection = owl( 'Hogwarts', 'Invitation' )
for i in range(100):
    collection = owl( collection, 'and another' )

Harry = collection # mark 2

print 'Uncle Vernon took his family and Harry to an isolated island'

Hagrid = 'motorcycle, hidden wand, cake'

if Dudley == "pig":
    Dudley = Dudley + ' tail'

Harry = 'Happy'
```

---

**Questions**

1. (4 points) What is the value of Harry at # mark 1?
2. (8 points) List all of the symbols in the global symbol table at the end of the code.
3. (4 points) Describe the value of the symbol `collection` at # mark 2. Do not write it all out.
4. (4 points) What is the value of the symbol `Dudley` at the end of the code?

**Name:**

---

### Python Code

```
def hat( house, student ) :
    house.append(student)

students = [ 'Ron', 'Hermione', 'Harry', 'Neville', 'Luna', 'Draco', 'Justin' ]
teachers = [ 'Hagrid', 'McGonagall', 'Sprout', 'Flitwick', 'Snape' ]

Gringotts = [ 'gold', 'gold', 'more gold' ]
Ollivanders = [ 'vine wood and dragon heartstring', 'holly and phoenix feather' ]
FlourishAndBotts = [ 'A History of Magic', 'Magical Drafts and Potions' ]

DiagonAlley = [ Gringotts, Ollivanders, FlourishAndBotts ]

print teachers[0] + ' took ' + students[2] + ' to Diagon Alley.'

print students[2] + ' bought ' + DiagonAlley[-1][-1] + ' with ' + DiagonAlley[0][1]

print students[1] + ' bought a ' + DiagonAlley[1][0] + ' wand.'

Gryffindor = [ teachers[1] ]
Ravenclaw = [ teachers[3] ]
Hufflepuff = [ teachers[-3] ]
Slytherin = [ teachers[4] ]

for firstyear in students[0:4]:
    hat( Gryffindor, firstyear )

hat( Hufflepuff, students[-1] )
hat( Ravenclaw, students[4] )
hat( Slytherin, students[-2] )
```

---

### Questions

1. (4 points) What does the first print statement write to the terminal?
2. (4 points) What does the second print statement write to the terminal?
3. (4 points) What does the third print statement write to the terminal?
4. (8 points) Write the values of Gryffindor, Hufflepuff, Ravenclaw, and Slytherin at the end of the code.

**Name:**

---

### Python Code

```
def Chess( kids ):
    finishers = []
    for kid in kids:
        if kid != 'Ron':
            finishers.append( kid )
    return finishers

def Hermione( set ):
    return set.pop()

def MagicalChallenges( kids ):

    devilsSnare = [kids[0], kids[2]]
    Hermione( devilsSnare )
    Hermione( devilsSnare )
    # mark 1

    keys = [ 'fake', 'fake', 'fake', 'fake', 'key' ]
    kids.append( Hermione( keys ) )

    kids = Chess( kids )
    # mark 2
    potions = [ 'paralyze', 'hot', 'cold' ]
    Harry = Hermione( potions )

    kids = kids[:1] + [ 'stone' ]
    LastRoom = ['Quirrel', 'Valdemort', kids[0]]

    print LastRoom[2] + ' finds the ' + kids[1] + ' and ' + LastRoom[1] + ' flees.'

if __name__ == "__main__":
    students = [ 'Harry', 'Hermione', 'Ron' ]
    MagicalChallenges( students )
```

---

### Questions

1. (5 points) What is the value of `devilsSnare` at mark 1?
2. (5 points) What is the value of `kids` at mark 2?
3. (5 points) What does the function `MagicalChallenges` print, given the code?
4. (5 points) What is the value of `kids` at the end of the `MagicalChallenges`?

Name:

---

### Python Code

```
def DiagonAlley( shoppers ):  
    shoppers[2].append( 'Diary' )  
    shoppers[3].append( 'Magical Me' )  
  
def HeadingToHogwarts( kids ):  
    kids[1][1] = 'broken wand'  
    return kids[0:2]  
  
def HogwartsFall( friends ):  
    badstuff = [ [ 'Ms. Norris', 'water' ], [ 'Colin', 'camera' ],  
                [ 'Justin', 'ghost' ], [ 'Nick', 'ghost' ] ]  
    friends[0].append( friends[2].pop() )  
    friends[-1].append( 'mirror' )  
    friends[-1].append( 'note' )  
    badstuff.append( friends.pop() )  
    friends[0].append( badstuff[-1].pop() )  
    return badstuff  
  
def ChamberOfSecretsPart1():  
    students = [ [ 'Harry', 'wand' ], [ 'Ron', 'wand' ],  
                [ 'Ginny', 'wand' ], [ 'Hermione', 'wand' ] ]  
  
    DiagonAlley( students )  
    print 'Lucius gives a ' + students[2][2] + ' to ' + students[-2][0]  
  
    car = HeadingToHogwarts( students )  
    print car[1][0] + ' ends the trip with a ' + car[1][-1]  
  
    petrified = HogwartsFall( students )  
    print students[0][0] + ' has a ' + students[0][2] + ' and ' + students[0][-1]  
    print petrified[-1][0] + ' cannot help.'  
  
ChamberOfSecretsPart1()
```

---

### Questions

1. (5 points) What does the first print statement write?
2. (5 points) What does the second print statement write?
3. (5 points) What does the third print statement write?
4. (5 points) What does the fourth print statement write?

**Name:**

---

**Python Code**

```
def CaveIn( RonsWand ):  
    for i in range( len(RonsWand) ):  
        RonsWand.pop()  
  
def Dumbledore( somehelp ):  
    items = []  
    for ch in somehelp:  
        if ch == 'c':  
            items.append( 'Faux' )  
        elif ch == 'r':  
            items.append( 'Sorting Hat' )  
        elif ch == 'g':  
            items.append( 'Sword of Gryffindor' )  
        elif ch == 'b':  
            items.append( 'Basilisk tooth' )  
    return items  
  
def Faux( person ):  
    person[-1].pop()  
  
def ChamberOfSecrets():  
    Lockhart = [ 'Many', 'magical', 'memories' ]  
    CaveIn( Lockhart )  
  
    Harry = Dumbledore( 'courage' )  
    Harry.pop()  
    Harry.append( ['Basilisk', 'tooth', 'poison'] )  
  
    TomRiddle = 80  
    for health in range( 60, -1, -20 ):  
        TomRiddle = health  
    Faux( Harry ) # mark 1  
  
ChamberOfSecrets() # mark 2
```

---

**Questions**

1. (8 points) Show the ChamberOfSecrets symbol table at #mark 1 (only name, type).
2. (4 points) Show the global symbol table at # mark 2 (only name, type).
3. (6 points) What are the values of the variables Lockhart and Harry at # mark 1?
4. (2 points) What is the value of the variable TomRiddle at # mark 1?

**Name:**

---

**Python Code**

```
def ToHogwarts( student ):  
    trip = student + 'BlackDog-KnightBus-DiagonAlley-'  
    trip += 'Train-Dementor'  
    return trip  
  
def Quidditch( visitors ):  
    guests = visitors[0]  
    for i in range(3):  
        guests = guests.replace( visitors[1][0], visitors[1][1] )  
    return guests  
  
def FredAndGeorge():  
    return '-MaraudersMap-BadAdvice'  
  
def Hogsmeade( visitor ):  
    return visitor + '-SiriusBlackRumors\n'  
  
def PrisonerOfAzkabanPart1():  
    Harry = ToHogwarts( '' )  
    uninvited = Harry[-8:]  
    # mark 1  
    unwelcome = Quidditch( [ uninvited, [ uninvited, 'Dementor-Dementor' ] ] )  
    print unwelcome # mark 2  
  
    Harry += FredAndGeorge()  
  
    if( Harry.find( 'Map' ) >= 0 ):  
        Harry = Hogsmeade( Harry )  
  
    fp = file( 'Holidays', 'w' )  
    fp.write( Harry )  
    fp.close() # mark 3  
  
PrisonerOfAzkabanPart1()
```

---

**Questions**

1. (4 points) What is the value of the variable uninvited at mark 1?
2. (4 points) What does the program print at mark 2?
3. (8 points) Write down the symbol table (name, type) for PrisonerOfAzkabanPart1 at mark 3.
4. (4 points) What is written to the file Holidays?

**Name:**

---

**Python Code**

```
class Wizard:
    def __init__( self, name, secrets ):
        self.name = name
        self.secrets = secrets

    def tellSecrets(self):
        return self.secrets

    def newSecret(self, newSecret):
        self.secrets = self.secrets + ', ' + newSecret

def PrisonerOfAzkabanPart2():
    Hermione = Wizard( 'Hermione', 'Hermione has a time machine' )
    Lupin = Wizard( 'Lupin', 'Lupin is a werewolf, friend of Sirius' )
    Snape = Wizard( 'Snape', 'Makes a potion for Lupin' )
    Scabbers = Wizard( 'Peter', 'Scabbers is really a person' )
    Sirius = Wizard( 'Sirius', 'Sirius is innocent' )
    Harry = Wizard( 'Harry', 'invisibility cloak' )
    # mark 1

    Sirius.newSecret( Scabbers.tellSecrets() )
    Lupin.newSecret( Snape.tellSecrets() )
    Lupin.newSecret( 'Pettigrew is alive' )
    Harry.newSecret( Lupin.tellSecrets() )
    Harry.newSecret( Sirius.tellSecrets() )

    Harry.newSecret( Hermione.tellSecrets() )
    Hermione.secrets = Harry.secrets # mark 2

    Dumbledore = Wizard( 'Albus', Harry.tellSecrets() )
    Ron = Wizard( 'Ron', Hermione.tellSecrets() )
    # mark 3

PrisonerOfAzkabanPart2()
```

---

**Questions**

1. (5 points) Write down the symbol table (name, type, value) for the object referenced by `Harry` at mark 1?
2. (5 points) Write down the `PrisonerOfAzkabanPart2` symbol table (name, type) at mark 3?
3. (5 points) Explain what the line of code at mark 2 does by referring to symbol tables.
4. (5 points) Which wizard objects contain all of the secrets?



**Name:**

---

### Python Code

```
class Character:
    def __init__(self, name ):
        self.name = name
        self.items = []

    def modify(self, item, mods):
        item.powers[mods[0]] = mods[1]

class MagicItem:
    def __init__(self, name, powers={}):
        self.name = name
        self.powers = powers.copy()

    def clone(self):
        return MagicItem(self.name, self.powers)

def FirstContest( contestants, goal ):
    for contestant in contestants:
        if contestant.name == 'Harry Potter':
            contestant.items.append( 'Broom' )
            contestant.items.append( goal.clone() )

def GobletOfFirePart1(goblet, Moody):
    Fleur = Character( 'Fleur Delacour' )
    Viktor = Character( 'Viktor Krum' )
    Cedric = Character( 'Cedric Diggory' )
    Harry = Character( 'Harry Potter' )

    Fleur.modify( goblet, ['Beauxbaton', Fleur] )
    Viktor.modify( goblet, ['Durmstrang', Viktor] )
    Cedric.modify( goblet, ['Hogwarts', Cedric] )
    Moody.modify( goblet, ['HogwartsB', Harry] ) # mark 1

    contestants = goblet.powers.values()
    egg = MagicItem( 'Golden Egg', {'message': 'rescue something in the lake'} )
    FirstContest( contestants, egg )

GobletOfFirePart1( MagicItem( 'Goblet of Fire' ), Character( 'Barty Crouch Jr.' ) )
```

---

### Questions

1. (5 points) At # mark 1, what are the names and types of each of the fields in the goblet object?
2. (5 points) Explain what the modify method does using # mark 1 as an example.
3. (5 points) At # mark 1, what are the key-value pairs in goblet.powers
4. (5 points) How many MagicItem objects exist at the end of GobletOfFirePart1? Think carefully.

Name:

---

### Python Code

```
def Action( contestant, lists, source, dest, action ):
    lists[dest].insert( 0, lists[source].pop() )
    print contestant, action, lists[dest][0]

def SecondTask( friends ):
    for i in range(3):
        friends['limbo'].append( friends['living'].pop() )
    # mark 1
    Action( 'Viktor', friends, 'limbo', 'living', 'saves' )
    Action( 'Harry', friends, 'limbo', 'living', 'saves' )
    Action( 'Harry', friends, 'limbo', 'living', 'saves' )

def Graveyard( victims ):
    Action( 'Valdemort', victims, 'living', 'dead', 'kills' )
    Action( 'Pettigrew', victims, 'limbo', 'living', 'restores' )
    Action( 'Harry', victims, 'dead', 'limbo', 'frees' )

def ThirdTask( actors ):
    Action( 'CrouchJr', actors, 'living', 'dead', 'kills' ) # mark 2
    Graveyard( actors )

def GobletOfFirePartII( people ):
    SecondTask( people )
    ThirdTask( people )
    Action( 'Dumbledore', actors, 'living', 'limbo', 'stuns' )
    Action( 'Dumbledore', actors, 'limbo', 'living', 'frees' )

actors = {}
actors['living'] = ['Dumbledore', 'Fleur', 'Viktor', 'Harry', 'CrouchJr',
                  'Cedric', 'CrouchSr', 'Hermione', 'Ron', 'Gabrielle', ]
actors['limbo'] = ['Moody', 'Valdemort']
actors['dead'] = ['former victims']
GobletOfFirePartII( actors )
```

---

### Questions

1. (6 points) Write out the value of the expression `friends['limbo']` at # mark 1.
2. (5 points) Explain what the function `Action` does, using the line right after # mark 1.
3. (5 points) What does the function `Action` print out at # mark 2.
4. (4 points) Write out the value of `people['dead']` at the end of `GobletOfFirePartII()`.

**Name:**

---

**Python Code**

```
def Umbridge( obstacle, action ):
    print 'Umbridge', action[0], obstacle[0]

    if len(obstacle) == 0 or obstacle[0] == 'centaurs':
        return 'Umbridge runs out of luck'

    return Umbridge( obstacle[1:], action[1:] )

def TheSetup(wizard, muggle, school):
    print 'Dementors attack', wizard, 'and', muggle
    print wizard, 'defends', muggle
    school[wizard] = False

    # mark 1
    if school['Dumbledore']:
        school[wizard] = True
        return 'Harry returns to Hogwarts'

def OrderOfThePhoenixPartI():
    Hogwarts = {'Harry':True, 'Ron':True, 'Hermione':True, 'Dumbledore':True }
    print TheSetup('Harry', 'Dudley', Hogwarts)

    events = ['Harry', 'Fred and George', 'D.A.',
              'Dumbledore', 'Harry, Ron, and Hermione', 'centaurs' ]
    actions = ['punishes', 'is frustrated by', 'discovers the',
              'replaces', 'captures', 'is captured by' ]

    # mark 2
    print Umbridge( events, actions )

OrderOfThePhoenixPartI()
```

---

**Questions**

1. (6 points) Write out the name, type, and value of the symbol table for TheSetup at # mark 1.
2. (5 points) Will the function call to Umbridge right after #mark 2 ever return? Explain your answer.
3. (4 points) How many times does the function Umbridge execute?
4. (5 points) What does the code print on the line after # mark 2?



**Code Sample 1**

```
def HalfBloodPrincePartI():
    prolog = 'Snape' + 'Draco'

    Teachers = [ 'McGonegal', 'Snape', 'Flitwick', 'Dumbledore' ]
    HarrysStuff = [ 'invisibilty cloak', 'marauders map' ]

    DumbledoresLuck = -1
    HarrysLuck = 2

    if DumbledoresLuck + HarrysLuck >= 1:
        Teachers.append( 'Slughorn' )

    HarrysLuck += 1
    HarrysStuff.append( 'annotated potions book' )

    HarrysLuck += 1
    print 'Harry wins the challenge from ' + Teachers[-1]
    HarrysStuff.append( 'liquid luck' )

    DumbledoresLuck -= 1
    HarrysStuff.append( 'Dumbledores memories' )

    HarrysStuff.pop(-2)
    HarrysLuck += 100
    HarrysStuff.append( 'Slughorns memories' )

    HarrysLuck -= 101
    # mark 1

HalfBloodPrincePartI()
```

**Code Sample 2**

```
def ThePedestal(actors, stuff, health):
    pedestal = ['locket', 'potion']
    stuff[ actors[0] ] = pedestal.pop()
    stuff[ actors[1] ] = pedestal.pop()
    health[ actors[0] ] -= 15

def GettingOut(actors, stuff, health):

    HarrysSpell = 10
    DumbledoresSpell = 90

    inferi = 100
    inferi -= HarrysSpell
    inferi -= DumbledoresSpell

    health[ actors[0] ] -= 5
    health[ actors[1] ] -= 5
    # mark 1

def TheCave(things, health):

    door = health['Dumbledore']/6
    health['Dumbledore'] -= 5

    boat = ['Dumbledore', 'Harry']
    ThePedestal( boat, things, health )

    GettingOut( boat, things, health )

def Hogwarts( things, health ):
    Draco = 0
    Snape = 5
    health[ 'Dumbledore' ] -= Draco
    health[ 'Dumbledore' ] -= Snape
    things.pop( 'Dumbledore' )
    health.pop( 'Dumbledore' )

def HalfBloodPrincePartII():
    actors = { 'Harry': None, 'Dumbledore' : None }
    health = { 'Harry': 100, 'Dumbledore' : 30 }

    TheCave( actors, health )

    Hogwarts( actors, health )

    print actors
    print health

# mark 2
HalfBloodPrincePartII()
```

**Code Sample 3**

```
def MinistryOfMagic( infiltrators):
    Umbridge = [ 'locket', '' ]
    fooled = 0
    for i in range(100):
        fooled += 1

    s = ''
    for p in infiltrators[:-1]:
        s = s + p + ', '
    s = s + 'and ' + infiltrators[-1]
    print s + ' steal the ' + Umbridge[0] # mark 1

    return Umbridge

def Gringotts( infiltrators ):
    cavern = [ 'gold', 'plate', 'cup', 'spoon' ]
    for person in infiltrators:
        if person == 'Harry':
            goal = 'cup'

    for item in cavern:
        if item == goal:
            return [ item, '' ]

def Hogwarts( infiltrators, opponents ):
    RoomOfRequirements = { 'pile 1': 'books', 'pile 2': 'diadem',
                          'pile 10014': 'cage' }
    for memory in infiltrators:
        if memory == 'Harry':
            location = 'pile 2'

    foundObject = [ RoomOfRequirements[ location ], '' ]
    foundObject[1] = 'fire'
    opponents.pop()
    # mark 2
    return foundObject

def DeathlyHallowsPartI():
    horcruxes = [ [ 'diary', 'basilisk tooth' ], [ 'ring', 'sword' ] ]
    hallows = [ 'cloak', 'wand', 'stone' ]
    students = [ 'Ron', 'Hermione', 'Harry' ]

    horcruxes.append( MinistryOfMagic( students ) )
    horcruxes[-1][1] = 'sword'

    horcruxes.append( Gringotts( students ) )
    horcruxes[-1][1] = 'tooth'

    horcruxes.append( Hogwarts( students, ['Malfoy', 'Goyle', 'Crabbe'] ) )
    print 'known horcruxes: ', horcruxes # mark 3

DeathlyHallowsPartI()
```

**Code Sample 4**

```
import random

class Character:
    def __init__(self, name):
        self.name = name
        self.health = 10

    def curse(self):
        return -1

    def aveda(self):
        return -10

    def attack(self):
        return -20

def FirstWave(good, evil):
    for combatant in good:
        target = random.choice( evil )
        target.health += combatant.curse()

    for combatant in evil:
        target = random.choice( good )
        target.health += combatant.curse()
    # mark 1

    good[3].health += random.choice(evil).aveda()
    good[6].health += random.choice(evil).aveda()
    good[7].health += random.choice(evil).aveda()
    evil[4].health += evil[5].attack()

def Woods( Harry, Voldemort ):
    Harry.health += Voldemort.aveda()
    Harry.health = 'limbo'
    # mark 2

def FinalDuel( good, evil ):
    good[0].health = 8
    evil[5].health += good[-2].attack()
    for i in range(10):
        evil[1].health += good[5].curse()
    evil[0].health += evil[0].aveda() * good[0].health
```



```
def DeathlyHallowsPartII():

    defenders = [ 'Harry', 'Ron', 'Hermione', 'Fred',
                  'George', 'Molly', 'Remus', 'Nymphadora',
                  'Neville', 'Luna' ]
    deatheaters = [ 'Valdemort', 'Bellatrix', 'Lucius',
                   'Greyback', 'Snape', 'Nagini', 'lots more' ]

    good = []
    for name in defenders:
        good.append( Character( name ) )
    evil = []
    for name in deatheaters:
        evil.append( Character( name ) )

    FirstWave( good, evil )
    Woods( good[0], evil[0] )
    FinalDuel( good, evil )
    # mark 3

    for agent in evil:
        if agent.health <= 0:
            print agent.name + ' is dead'
    for agent in good:
        if agent.health <= 0:
            print agent.name + ' is dead'

    if good[0].health > 0:
        print good[0].name + ' is alive'

DeathlyHallowsPartII()
```

**Code Sample 5**

```
class Person:
    def __init__(self, name):
        self.name = name
        self.happiness = 10

class Parent( Person ):
    def __init__(self, name, house):
        Person.__init__(self, name)
        self.house = house

class Kid( Person ):
    def __init__(self, name, parents):
        Person.__init__(self, name)
        self.parents = []
        for p in parents:
            self.parents.append( p )

    def hug( self, parent ):
        self.happiness += 1
        parent.happiness += 1

def DeathlyHallowsEpilog():
    Harry = Parent( 'Harry', 'Gryffindor' )
    Ginny = Parent( 'Ginny', 'Gryffindor' )
    James = Kid( 'James', [Harry, Ginny] )
    Albus = Kid( 'Albus', [Harry, Ginny] )
    Lily = Kid( 'Lily', [Harry, Ginny] )
    Albus.hug( Ginny )
    Albus.hug( Harry )

    Ron = Parent( 'Ron', 'Gryffindor' )
    Hermione = Parent( 'Hermione', 'Gryffindor' )
    Rose = Kid( 'Rose', [Ron, Hermione] )
    Hugo = Kid( 'Hugo', [Ron, Hermione] )
    Rose.hug( Ron )
    Rose.hug( Hermione )

    Draco = Parent( 'Draco', 'Slytherin' )
    Astoria = Parent( 'Astoria', 'Slytherin' )
    Scorpius = Kid( 'Scorpius', [Draco, Astoria] )
    Scorpius.hug( Astoria )
    Scorpius.hug( Draco )
    # mark 1

DeathlyHallowsEpilog()
```

**Name:**

---

### **Code Sample 1 Questions**

1. (20 points) Write out the all of the symbol tables—including name, type, and value—that exist at mark 1 during program execution.

**Code Sample 2 Questions**

1. (5 points) What is the relationship between the `stuff` parameter in `ThePedestal` and the `actors` variable in `HalfBloodPrincePartII`?
2. (6 points) Write out the `GettingOut` symbol table—with name, type, and value—at mark 1.
3. (5 points) Write out the global symbol table—with name and type—at mark 2.
4. (4 points) Write out what the function `HalfBloodPrincePartII` prints.

**Code Sample 3 Questions**

1. (6 points) Write out the Hogwarts symbol table—including name, type, and value—at mark 2.
2. (4 points) What would be the return value of the Gringotts function if the string 'cup' was not in the cavern list?
3. (5 points) What does the function MinistryOfMagic print at mark 1?
4. (5 points) What does the function DeathlyHallowsPartI print at mark 3?

**Code Sample 4 Questions**

1. (5 points) How many `Character` objects exist at `mark 3` in `DeathlyHallowsPartII`?
2. (4 points) Write out the symbol tables—name, type, and value—for the variables `Harry` and `Valdemort` at `mark 2` in the `Woods` function.
3. (3 points) What is the type and value of the `combatant` variable at `mark 1` in the `FirstWave` function?
4. (8 points) What does the function `DeathlyHallowsPartII` print?

**Code Sample 5 Questions**

1. (9 points) Write out the symbol tables—including name and type—for Person, Parent, and Kid.
2. (4 points) Write out the symbol table—including name, type, and value—for Harry at mark 1.
3. (4 points) Write out the symbol table—including name, type, and value—for Albus at mark 1.
4. (3 points) What is the type and value of the expression `Harry.happiness` at mark 1?