Contractor

constructor with inputs (“bob”, “Library”, “Harold”). toString() expects “bob Library Harold”

```
> Contractor c = new Contractor("bob", "Library", "Harold");
> c.toString()
"bob Library Harold"
```

equals(Object o) expects true

```
> Contractor c = new Contractor("bob", "Library", "Harold")
> Contractor c2 = new Contractor("bob", "Library", "adsf")
> c.equals(c2)
true
```

equals(Object o) expects false

```
> c2.setName("Bob")
> c.equals(c2)
false
> c.equals("AHA")
false
> c.equals(d)
false                    //d is an instance of Date
```

setName(“Bob”) and getName(). Expects “Bob”

```
> c.setName("Bob")
> c.getName()
"Bob"
```

setAddress(“Edison”) and getAddress(). Expects “Edison”

```
> c.setAddress("Edison")
> c.getAddress()
"Edison"
```

setContact(“Edison”) and getContact(). Expects “Edison”

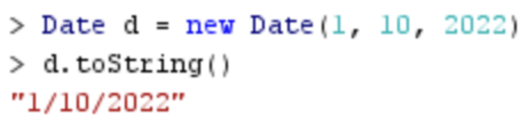
```
> c.setContact("Edison")
> c.getContact()
"Edison"
```

pay(20.0). getAmoutPaid() expects 20.0. pay(20.0). getAmountPaid() expects 40.0

```
> c.pay(20.0)
> c.getAmountPaid()
20.0
> c.pay(20.0)
> c.getAmountPaid()
40.0
```

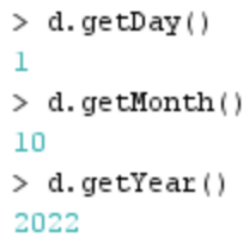
Date

constructor with inputs (1, 10, 2022). toString() expects “1/10/2022”



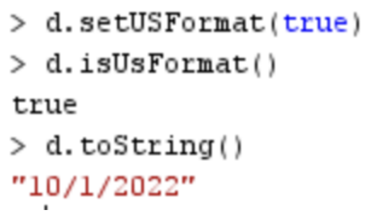
```
> Date d = new Date(1, 10, 2022)
> d.toString()
"1/10/2022"
```

getDay() expects 1. getMonth() expects 10. getYear() expects 2022



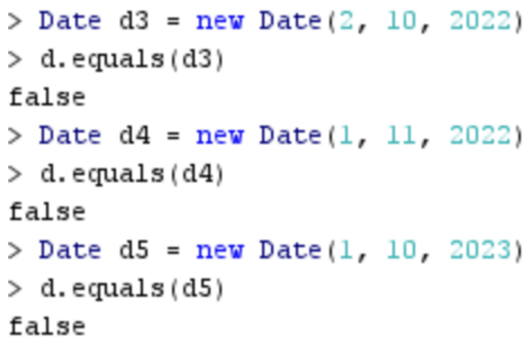
```
> d.getDay()
1
> d.getMonth()
10
> d.getYear()
2022
```

setUSFormat(true). isUsFormat() expects true. toString() expects “10/1/2022”

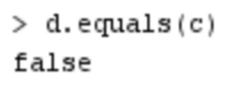


```
> d.setUSFormat(true)
> d.isUsFormat()
true
> d.toString()
"10/1/2022"
```

equals(Object o) expects false



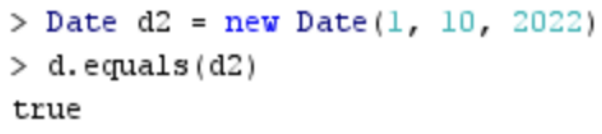
```
> Date d3 = new Date(2, 10, 2022)
> d.equals(d3)
false
> Date d4 = new Date(1, 11, 2022)
> d.equals(d4)
false
> Date d5 = new Date(1, 10, 2023)
> d.equals(d5)
false
```



```
> d.equals(c)
false
```

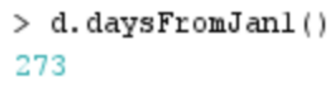
//c is an instance of Contractor

equals(Object o) expects true



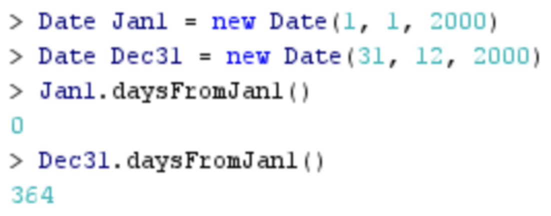
```
> Date d2 = new Date(1, 10, 2022)
> d.equals(d2)
true
```

daysFromJan1() expects 273



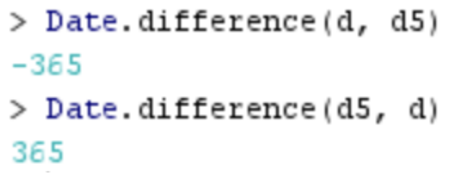
```
> d.daysFromJan1()
273
```

daysFromJan1() expects 0. daysFromJan1() expects 364



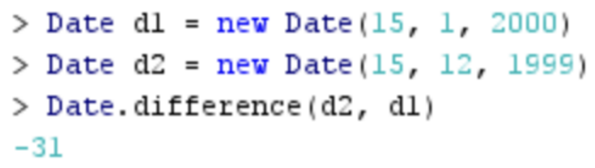
```
> Date Jan1 = new Date(1, 1, 2000)
> Date Dec31 = new Date(31, 12, 2000)
> Jan1.daysFromJan1()
0
> Dec31.daysFromJan1()
364
```

difference(d, d5) expects -365. (d is Oct 1, 2022; d5 is Oct 1 2023)



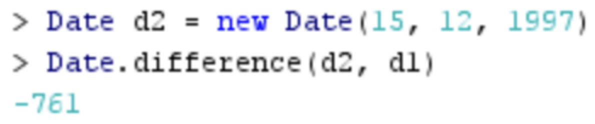
```
> Date.difference(d, d5)
-365
> Date.difference(d5, d)
365
```

difference(d2, d1) expects -31



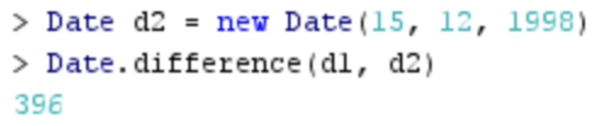
```
> Date d1 = new Date(15, 1, 2000)
> Date d2 = new Date(15, 12, 1999)
> Date.difference(d2, d1)
-31
```

difference(d2, d1) expects -761



```
> Date d2 = new Date(15, 12, 1997)
> Date.difference(d2, d1)
-761
```

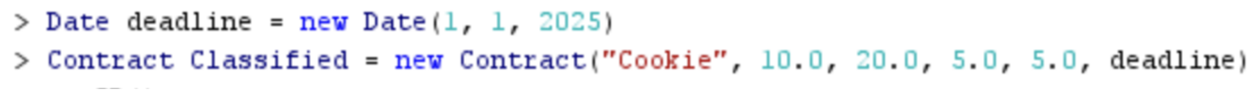
difference(d1, d2) expects 396



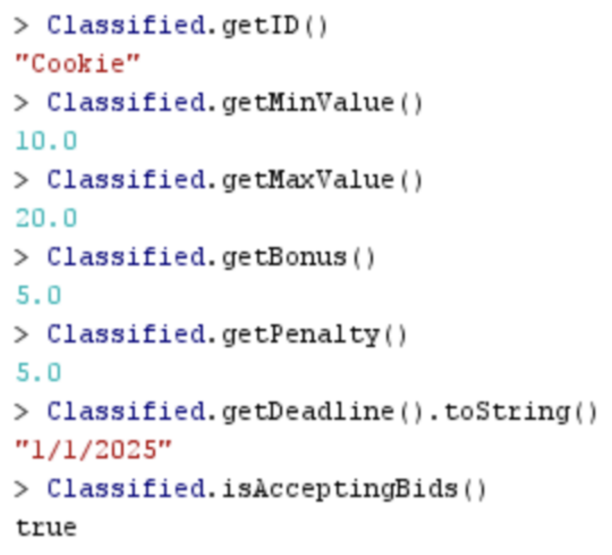
```
> Date d2 = new Date(15, 12, 1998)
> Date.difference(d1, d2)
396
```

Contract

constructor with inputs (“Cookie”, 10.0, 20.0, 5.0, 5.0, deadline)

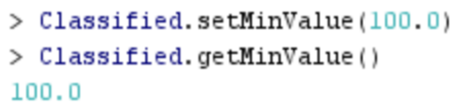


```
> Date deadline = new Date(1, 1, 2025)
> Contract Classified = new Contract("Cookie", 10.0, 20.0, 5.0, 5.0, deadline)
```



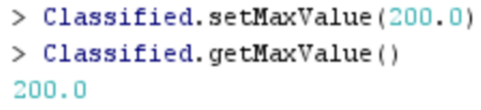
getID() expects “Cookie”	`> Classified.getID()` `"Cookie"`
getMinValue() expects 10.0	`> Classified.getMinValue()` `10.0`
getMaxValue() expects 20.0	`> Classified.getMaxValue()` `20.0`
getBonus() expects 5.0	`> Classified.getBonus()` `5.0`
getPenalty() expects 5.0	`> Classified.getPenalty()` `5.0`
getDeadline().toString() expects “1/1/2025”	`> Classified.getDeadline().toString()` `"1/1/2025"`
isAcceptingBids() expects true	`> Classified.isAcceptingBids()` `true`

setMinValue(100.0). getMinValue() expects 100.0



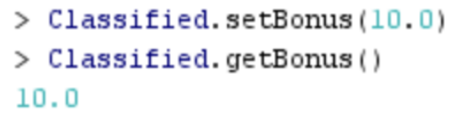
```
> Classified.setMinValue(100.0)
> Classified.getMinValue()
100.0
```

setMaxValue(200.0). getMaxValue() expects 200.0



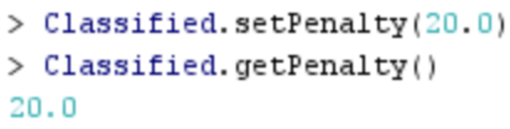
```
> Classified.setMaxValue(200.0)
> Classified.getMaxValue()
200.0
```

setBonus(10.0). getBonus() expects 10.0



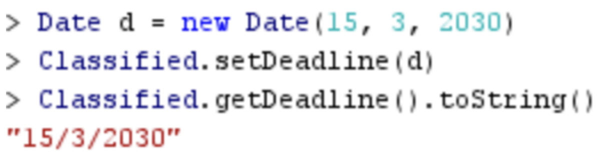
```
> Classified.setBonus(10.0)
> Classified.getBonus()
10.0
```

setPenalty(20.0). getPenalty() expects 20.0



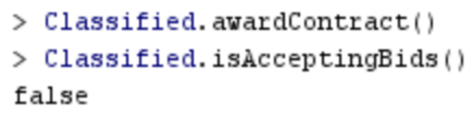
```
> Classified.setPenalty(20.0)
> Classified.getPenalty()
20.0
```

setDeadline(d). getDeadline.toString() expects “15/3/2030”



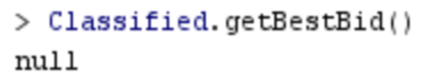
```
> Date d = new Date(15, 3, 2030)
> Classified.setDeadline(d)
> Classified.getDeadline().toString()
"15/3/2030"
```

awardContract(). isAcceptingBids() expects false



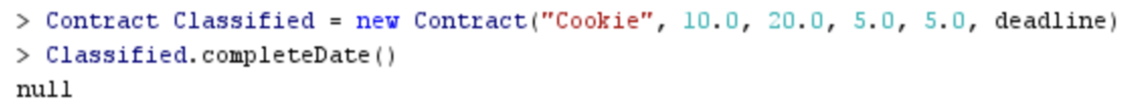
```
> Classified.awardContract()
> Classified.isAcceptingBids()
false
```

getBestBid() expects null



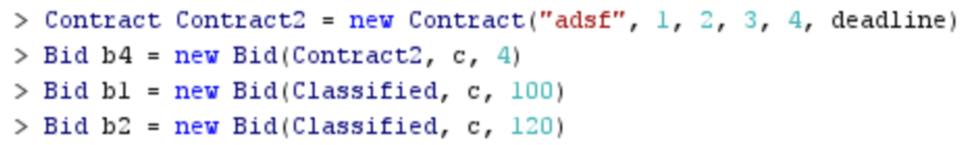
```
> Classified.getBestBid()
null
```

completeDate() expects null

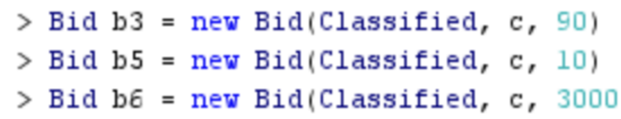


```
> Contract Classified = new Contract("Cookie", 10.0, 20.0, 5.0, 5.0, deadline)
> Classified.completeDate()
null
```

Testing makeBid(Bid bid)

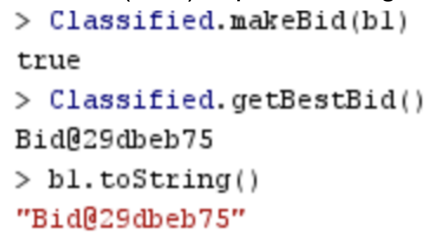


```
> Contract Contract2 = new Contract("adsf", 1, 2, 3, 4, deadline)
> Bid b4 = new Bid(Contract2, c, 4)
> Bid b1 = new Bid(Classified, c, 100)
> Bid b2 = new Bid(Classified, c, 120)
```



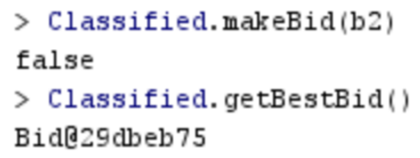
```
> Bid b3 = new Bid(Classified, c, 90)
> Bid b5 = new Bid(Classified, c, 10)
> Bid b6 = new Bid(Classified, c, 3000)
```

makeBid(bid1) expects true. getBestBid() expects bid1



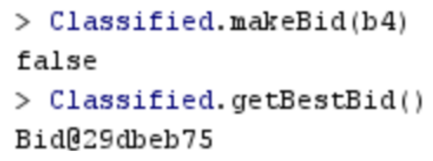
```
> Classified.makeBid(b1)
true
> Classified.getBestBid()
Bid@29dbeb75
> b1.toString()
"Bid@29dbeb75"
```

makeBid(bid2) expects false. getBestBid() expects bid1



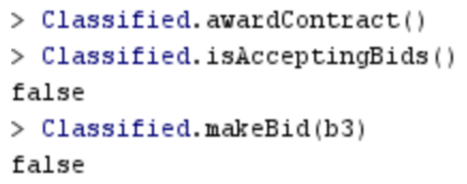
```
> Classified.makeBid(b2)
false
> Classified.getBestBid()
Bid@29dbeb75
```

makeBid(bid4) expects false. getBestBid() expects bid1



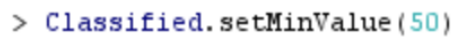
```
> Classified.makeBid(b4)
false
> Classified.getBestBid()
Bid@29dbeb75
```

makeBid(b3) expects false. No longer accepting bids

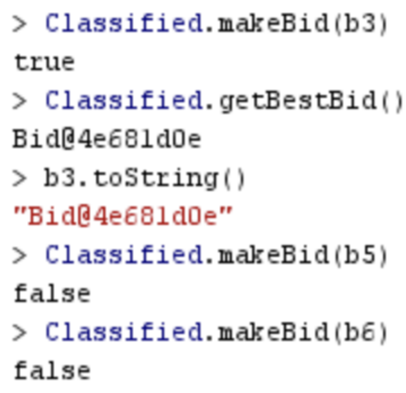


```
> Classified.awardContract()
> Classified.isAcceptingBids()
false
> Classified.makeBid(b3)
false
```

makeBid(b3) expects true. getbestBid() expects bid3. makeBid(b5) expects false.
makeBid(b6) expects false

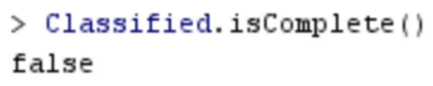


```
> Classified.setMinValue(50)
```



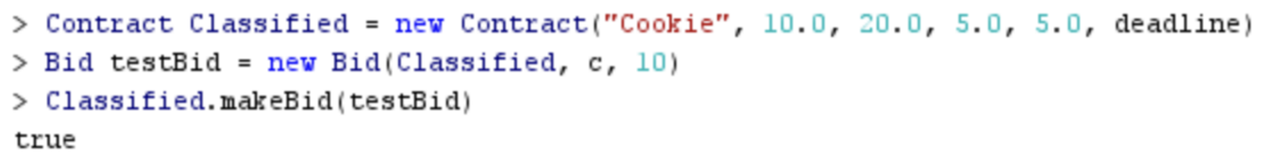
```
> Classified.makeBid(b3)
true
> Classified.getBestBid()
Bid@4e681d0e
> b3.toString()
"Bid@4e681d0e"
> Classified.makeBid(b5)
false
> Classified.makeBid(b6)
false
```

isComplete() expects false



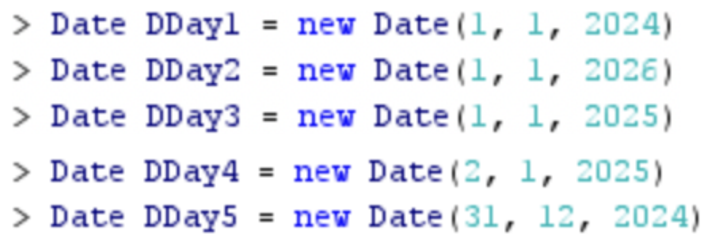
```
> Classified.isComplete()
false
```

Test setComplete(Date date) //Crashes if there is no bid made



```
> Contract Classified = new Contract("Cookie", 10.0, 20.0, 5.0, 5.0, deadline)
> Bid testBid = new Bid(Classified, c, 10)
> Classified.makeBid(testBid)
true
```

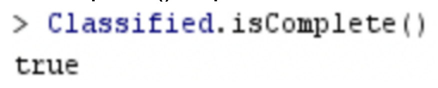
//deadline is 1/1/2025



```
> Date DDay1 = new Date(1, 1, 2024)
> Date DDay2 = new Date(1, 1, 2026)
> Date DDay3 = new Date(1, 1, 2025)
> Date DDay4 = new Date(2, 1, 2025)
> Date DDay5 = new Date(31, 12, 2024)
```

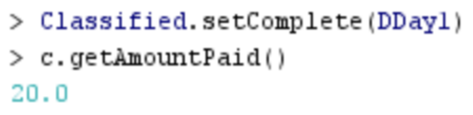
Contractor c.getAmountPaid is always reset to 0.

setComplete(DDay1) //bonus too much
isComplete() expects true



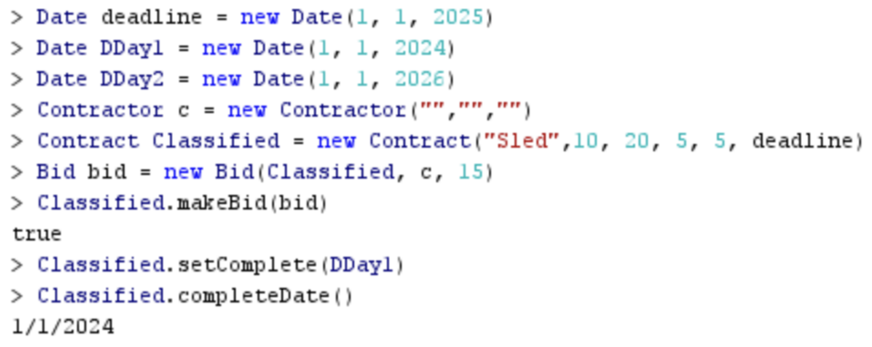
```
> Classified.isComplete()
true
```

c.getAmountPaid() expects maxValue = 20.0



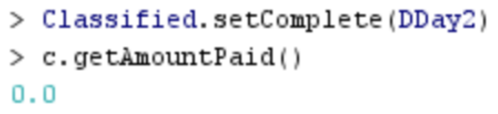
```
> Classified.setComplete(DDay1)
> c.getAmountPaid()
20.0
```

completeDate() expects 1/1/2024



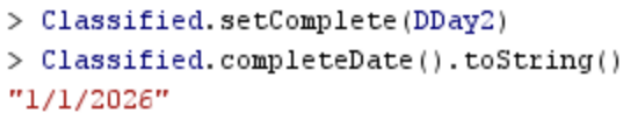
```
> Date deadline = new Date(1, 1, 2025)
> Date DDay1 = new Date(1, 1, 2024)
> Date DDay2 = new Date(1, 1, 2026)
> Contractor c = new Contractor("","","")
> Contract Classified = new Contract("Sled",10, 20, 5, 5, deadline)
> Bid bid = new Bid(Classified, c, 15)
> Classified.makeBid(bid)
true
> Classified.setComplete(DDay1)
> Classified.completeDate()
1/1/2024
```

setComplete(DDay2) //penalty too much
c.getAmountPaid() expects 0.0



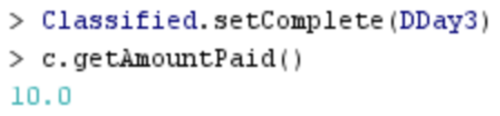
```
> Classified.setComplete(DDay2)
> c.getAmountPaid()
0.0
```

completeDate().toString() expects “1/1/2026”



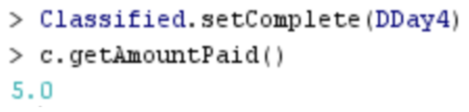
```
> Classified.setComplete(DDay2)
> Classified.completeDate().toString()
"1/1/2026"
```

setComplete(DDay3)
c.getAmountPaid() expects bestBid.value() = 10.0



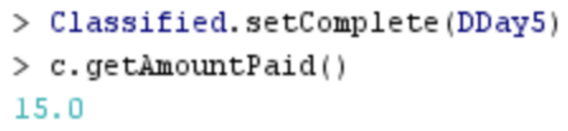
```
> Classified.setComplete(DDay3)
> c.getAmountPaid()
10.0
```

setComplete(DDay4)
c.getAmountPaid() expects (10 - 5*(1) = 5.0)



```
> Classified.setComplete(DDay4)
> c.getAmountPaid()
5.0
```

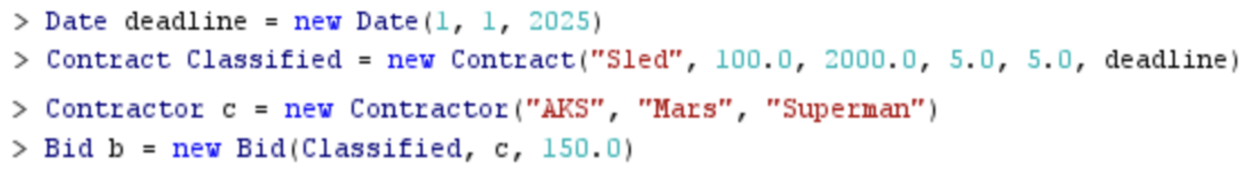
setComplete(DDay5)
c.getAmountPaid() expects (10 - 5*(-1) = 15.0)



```
> Classified.setComplete(DDay5)
> c.getAmountPaid()
15.0
```

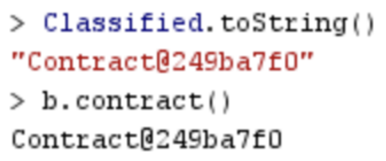
Bid

constructor with inputs(Classified, c, 150.0)



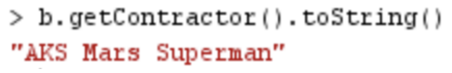
```
> Date deadline = new Date(1, 1, 2025)
> Contract Classified = new Contract("Sled", 100.0, 2000.0, 5.0, 5.0, deadline)
> Contractor c = new Contractor("AKS", "Mars", "Superman")
> Bid b = new Bid(Classified, c, 150.0)
```

contract() expects address



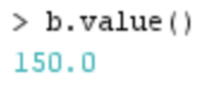
```
> Classified.toString()
"Contract@249ba7f0"
> b.contract()
Contract@249ba7f0
```

getContractor.toString() expects “AKS Mars Superman”



```
> b.getContractor().toString()
"AKS Mars Superman"
```

value() expects 150.0



```
> b.value()
150.0
```