632068
Note draft only topics and explanation tba

Consider the following short program:

```
#include <stdio.h>

void f1(void);

int a;

void main(void)

{

     int b;

     a = b = 1;

     f1();

     printf("%d %d", a, b);

}

void f1(void)

{

     int b = 3;

     a = b;

}
```

The output from the print statement will be:

a. 1 1

b. 1 3

*c. 3 1

d. 3 3

e. 3 5

f. "
g. "
h. "
i. "
j. "

General Feedback:

Explanation to be completed

633240
Finding the minimum value in a complete and balanced binary search tree is

a. O(1)

*b. O(log N)

c. O(N)

d. O(N^2)

e. O(N log N)

f. "
g. "
h. "
i. "
j. "

General Feedback:

The minimum is the root's left-most descendent, which can be reached in log N steps.

634971
An interface in Java can appropriately be used to model

a. Abstract classes

*b. Unrelated classes that share particular capabilities

c. Related classes that share particular attributes

d. Classes that communicate using the same network protocol

e. None of the above

f. "
g. "
h. "
i. "
j. "

General Feedback:

Interfaces allow us to specify common capabilities in unrelated classes: a way to say that two classes have the same methods without specifying the implementation (and allowing completely different implementations). For closely related classes (near each other in inheritance hierarchy) we can get common capabilities by having them in a common ancestor. Abstract classes provide a method to specify common capabilities via inheritance. Interfaces cannot be used to specify that two classes share a set of attributes (i.e. instance variables), which can be done by having classes inherit from a common ancestor with those attributes.