

QUEEN MARY, UNIVERSITY OF LONDON

DCS128 ALGORITHMS AND DATA STRUCTURES

Class Test Monday 13th February 2006 11.05-12.35

Please fill in your Examination Number here:

Student Number here:

All answers to this test should be written on the test sheet, but you may use spare paper for rough working. Answer as many of the questions as you can.

- 1) Suppose that in Java class `DrinksMachine` and `Can` there are public methods with the following signatures:

`DrinksMachine` methods:

```
void insert(int n)
int getBalance()
int getPrice()
boolean cokesEmpty()
Can pressCoke()
void loadCoke(Can can)
void setPrice(int p)
```

`Can` methods:

```
boolean isFull()
void drink()
int volume()
```

Suppose also we have variables `mach1` and `mach2` of type `DrinksMachine`, and `c1` and `c2` of type `Can`, and you can assume they have been initialised to refer to objects of the appropriate type, and suppose also we have variable `n` of type `int` which has been initialised.

For each of the code fragments below, state whether it is valid or invalid code, where invalid code is code which would cause a compiler error.

Valid/Invalid

- a) `n = mach1.getPrice();`
.....
- b) `n = mach2.setPrice(n);`
.....
- c) `c1 = mach1.pressCoke();`
`if(c1.isFull())`
`c2 = mach2.pressCoke();`
.....
- d) `if(insert(n))`
`c2 = mach2.pressCoke();`
.....
- e) `while(c1.volume()>0)`
`c1.drink();`
.....

[Question continued on next page]

- f)

```
while(mach1.cokesEmpty())
    mach1.loadCoke(mach2.pressCoke());
```


- g)

```
n = mach1.pressCoke().volume();
```


- h)

```
mach1.pressCoke();
if(c1.isFull())
    c2 = mach2.pressCoke();
```


- i)

```
if(mach1.getPrice()>n)
    mach1.setPrice() = n;
```


- j)

```
while(mach1.cokesEmpty())
{
    int p = mach2.getPrice();
    if(p>n)
        mach1.loadCoke(c1.volume());
}
```


Use the space below for rough working

- 2) a) Write a Java static method `slide` which takes as its argument an array of integers and alters it destructively so that the last integer is put in the first place, and all other integers are moved up one place. So if the array starts off as:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 7 | 9 | 3 | 7 | 4 | 1 | 2 | 8 | 7 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|

it becomes:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 5 | 7 | 9 | 3 | 7 | 4 | 1 | 2 | 8 | 7 |
|---|---|---|---|---|---|---|---|---|---|---|

- b) Write in the space below a Java static method which performs the same operation as in part a), but does it constructively.

- 3) a) Write a Java static method which takes an arrayList and an object and returns an arrayList containing all items from the original arrayList which are not equal to the argument object in the reverse order to the one they had in the original arrayList.

For example, if the original arrayList contains Integer values as follows:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 7 | 9 | 3 | 7 | 4 | 1 | 2 | 8 | 7 | 3 |
|---|---|---|---|---|---|---|---|---|---|---|

and the second argument is the Integer value 3, the arrayList that is returned is:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 7 | 8 | 2 | 1 | 4 | 7 | 9 | 7 | 5 |
|---|---|---|---|---|---|---|---|---|

Your method should be generic to cover arrayLists of all base types.

- b) Below is a Java method:

```
public static <T> int meth(ArrayList<T> a)
{
    int n=0
    for(int i=0; i<a.size(); i++)
    {
        T thing = a.get(i);
        int j=i+1;
        for(; j<a.size(); j++)
            if(a.get(j).equals(thing)
                break;
        if(j==a.size())
            n++;
    }
    return n;
}
```

State in English what value this method returns:

- 4) a) Describe briefly the difference between a **mutable** and an **immutable** object.
- b) Explain briefly what is meant by **aliasing** and why this may be a problem with mutable but not immutable objects.
- c) Explain what the `String` method `toUpperCase` does.

- d) Assuming `str1` is set to a string, and `ch` is set to a character which we know occurs at least twice in the string, write a fragment of code which sets variable `str2` to the portion of `str1` between the first and last occurrence of `ch`. So if `str1` is "hello world" and `ch` is 'o', it should set `str2` to "o wo".
- You may use the `String` methods `indexOf` which takes a character as its argument and returns the position of the first occurrence of that character in the string it is called on, and `lastIndexOf` which is similar, but it returns the position of the last occurrence of the character.

