

1. Suppose we have a line with 4 values representing a lab mark, assignment mark, test mark, and final exam mark; each mark is out of 100. For example "99.1 81.3 80.5 85.9".

There are other techniques that may be useful, but note that a Scanner object can be created for a string. Suppose `line` is of type `String`, then one can code:

```
Scanner s = new Scanner(line);
```

Write a method that calculates the final mark (out of 100) where labs contribute 10% toward the final mark, assignments contribute 20%, the test mark contributes 30% and the exam mark contributes 40%.

The method begins

```
public double finalMark(String line) {
```

2. Write a method that returns the grade for a given mark. You may assume that the mark will be in the range [0, 100].

This table relates marks to grades.

Grade	Mark range
A	80-100
B	70-79
C	60-69
D	50-59
F	0-49

Example: The statement `System.out.println(getGrade(76));` will display

B

The method begins

```
public String getGrade(int mark){
```

3. Suppose we have two ArrayLists: `correctAnswers` and `studentAnswers`.
`correctAnswers` holds the correct answers to a multiple-choice test.
`studentAnswers` contains a student's answers to the test.

The score for a test is the number of questions the student answered correctly. Given the sample data below,

```
correct:  a b c d a b c d a b c d
answers:  a a a b b b c c c d d d
```

The statement `System.out.println(score(correct, answers));` will display
4

Write the method `score` that returns a student's score for a test. Your code must work for any values in `correctAnswers` and `studentAnswers`. The method begins:

```
public int score (
    ArrayList<String> correctAnswers, ArrayList<String> studentAnswers)
{
```

4. Write a method `getCheckDigit` to determine the check digit for an **11-digit** Universal Product Code. The check digit is calculated as follows:
- Add the digits in the odd-numbered positions (first, third, fifth, etc.) together and then multiply by three.
 - Add the digits in the even-numbered positions (second, fourth, sixth, etc.) to the previous sum.
 - Calculate sum modulo 10 (i.e. the remainder when divided by 10)
 - If the result is not zero, subtract the result from ten.

For example, to obtain the check digit for 03600029145:

- adding the odd-numbered digits ($0 + 6 + 0 + 2 + 1 + 5 = 14$),
multiplying by three ($14 \times 3 = 42$),
- adding the even-numbered digits ($42 + (3 + 0 + 0 + 9 + 4) = 58$),
- calculating modulo ten ($58 \bmod 10 = 8$),
- subtracting from ten ($10 - 8 = 2$).

So, `System.out.println(getCheckDigit("03600029145"))`
displays

2

The method begins

```
public int getCheckDigit(String upc) {
```

7. Suppose an `ArrayList` contains integers. Complete the method below that returns an `ArrayList` containing all the numbers excluding the smallest one. You may assume all numbers in the list are different from each other.

For example, if

```
ArrayList<Integer> marks  
contains the values 5, 9, 3, 22, 8, 15, 12  
then
```

```
ArrayList<Integer> result = findHighest(marks);  
for (Integer r: result) System.out.println(r);  
displays:
```

```
5  
9  
22  
8  
15  
12
```

The method begins

```
public ArrayList<Integer> findHighest(ArrayList<Integer> m) {
```

8. Consider the Library, Member, and Driver classes below. The *main()* method in Driver (on next page) adds members to the library and then displays members who live in the city of “Winnipeg” and whose last name begins with an “S”. Complete the Member and Library classes so the sample *main()* method works correctly.

Do not change the *main()* method. Note that sample data is shown in the *main()* method, but your code must work regardless of the number of members, their field values, and regardless of the arguments passed to *displayMembersWho*.

```
/**
 * The library has members who take out books.
 */
public class Library
{
    // an arraylist of members
    private ArrayList<Member> libMembers;    /**
     * Constructor
     */
    public Library()
    {
        libMembers = new ArrayList<Member>();
    }

    /**
     * add a new member
     */
    public void addMember(Member m)
    {
        // put your code here
    }

    /**
     * display the members who live in "c" and
     * have a last name beginning with "f"
     */
    public void displayMembersWho(String c, String f)
    {
        // put your code here
    }
}
```

```
public class Member
{
    private int id;
    private String firstName;
    private String lastName;
    private String street;
    private String city;

    public Member (int i, String f, String l,String s, String c)
    {
        id          = i;
        firstName   = f;
        lastName    = l;
        street      = s;
        city        = c;
    }
    public String getLastName()
    { // put your code here

    }
    public String getFirstName()
    { // put your code here

    }
    public String getCity()
    { // put your code here

    }
    public int getId()
    { // put your code here

    }
}

public class Driver
{
    public static void main(String[] args){
        Library lib = new Library();
        // create some sample members for demonstration purposes
        Member t1 = new Member (111,"Joe","Smith","123 Portage Ave", "Winnipeg");
        Member t2 = new Member (112,"Jim","Jones","111 Victoria Ave", "Brandon");
        Member t3 = new Member (113,"Betty","Sanii","55 Main St", "Brandon");
        Member t4 = new Member (114,"April","Peters","3 Victoria Ave", "Brandon");
        Member t5 = new Member (115,"Dan","Stevens","24 Ellice Ave", "Winnipeg");
        // add members to the library
        lib.addMember (t1);
        lib.addMember (t2);
        lib.addMember (t3);
        lib.addMember (t4);
        lib.addMember (t5);
        // Who lives in Winnipeg and has a last name beginning with an S?
        lib.displayMembersWho ("Winnipeg", "S");
    }
}
```

The output from the **main()** method will be:

```
111 Joe Smith
115 Dan Stevens
```