CC-214-L: Object Oriented Programming Lab
BSSE Fall 2020
Lab 08

Issue Date/Time: 5-Dec-2021/10:00AM
Submission Date/Time: 5-Dec-2021/03:00PM
Marks: 43

Objective:

- To explore/implement an application of Composition relationship.

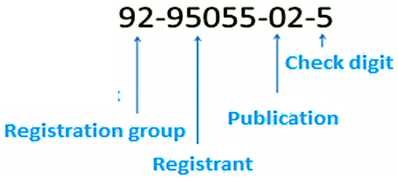
Challenge-A: ***ISBN10 and ISBN-13***

In the late 1960's, book publishers realized that they needed a uniform way to identify all the different books that were being published throughout the world. In 1970 they came up with the International Standard Book Number system. Every book, including new editions of older books, was to be given a special number, called an ISBN, which is not given to any other book.

It is to be noted that the ISBN is *10*-digits long if assigned before 2007, and *13*-digits long if assigned on or after 1 January 2007.

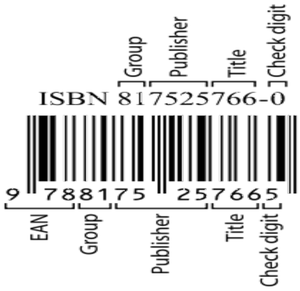
An ISBN is a 10 digit "structured" number - different parts of the number have different meanings (similar to the ZIP codes). The parts of the number are separated by spaces or hyphens (hyphens are preferred, but not required.) The ISBN is broken up into four parts, the sizes of the first three parts are variable, but the total number of digits used in these parts must add up to nine. The last digit is a check digit which is calculated from the previous nine digits. The ISBN's are usually printed on the back cover of a book and look like these examples:

ISBN 0-471-19047-0   **ISBN 87-11-07559-7**

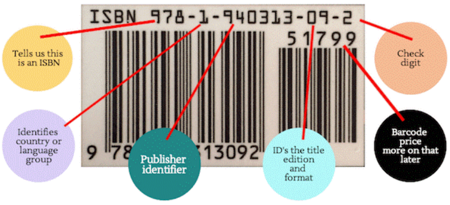


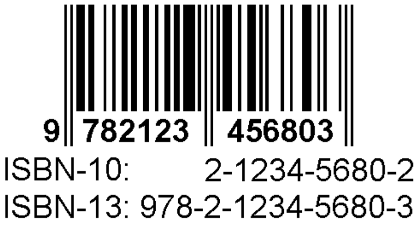
- The first part of the ISBN identifies a country, area or language area participating in the ISBN system.
  - Some members form language areas (e.g. group number 3 = German language group) or regional units (e.g. South Pacific = group number 982). A group identifier may consist of up to ***5*** digits. Group number 0 is an English language group which includes the United States, the United Kingdom, Australia, South Africa and other countries. Group number 87 is Denmark, group number 627 is Pakistan, and group number 99942 is Sudan (Africa).
- The second part of the ISBN identifies a particular publisher within a group.
  - The publisher identifier usually indicates the exact identification of the publishing house and its address. If publishers use up their initial collection of title numbers, they may be given an additional publisher identifier. The publisher identifier consists of up to ***7*** digits.
- The third part of the ISBN identifies a specific edition of a publication of a specific publisher. A title identifier may consist of up to ***6*** digits.
- The fourth digit is the check digit. It is always the final single digit that mathematically validates the rest of the number.

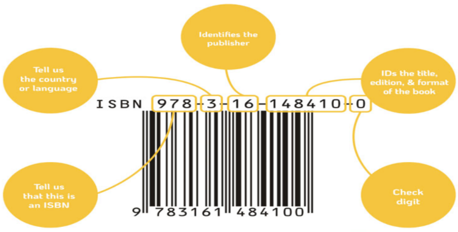
For 13-digit ISBN, a 3-digit prefix (978 or 979) is added before 10-digit code. This 3-digit prefix offers more to the book industry. ISBN-13 also conform with the European Article Number (EAN) barcode format found on most commercial merchandise.



Some Sample ISBN-13 and ISBN-10


ISBN 978-1-940313-09-2
51799
9 78 13092
Tells us this is an ISBN
Identifies country or language group
Publisher identifier
ID's the title edition and format
Check digit
Barcode price more on that later




Identifies the publisher
Tell us the country or language
IDs the title, edition, & format of the book
ISBN 978-3-16-148410-0
Tell us that this is an ISBN
Check digit
9 783161 484100

Hopefully, you know much about ISBN now but before we move onto coding stuff lets learn a bit about the purpose of check digit.

What The ISBN Check Digit Is For?
A check digit is a form of redundancy check used for error detection on identification numbers, such as bank account numbers, which are used in an application where they will at least sometimes be input manually. It consists of one or more digits (or letters) computed by an algorithm from the other digits (or letters) in the sequence input.
With a check digit, one can detect simple errors in the input of a series of characters (usually digits) such as a single mistyped digit or some permutations of two successive digits.

In the same way, the check digit in an ISBN number is there to help prevent errors in transmission. When an ISBN number is read and entered into any system that is used to deal with ISBNs, such as a book cataloguing system, whether the ISBN is entered by hand or by means of a bar code scanner, the check digit as read from the entered number is compared to a check digit calculated from the remainder of the ISBN that has been entered. If the two check digits are the same, then there is a good chance that the complete ISBN, all of its digits, has been read correctly.

ISBN-10 check digit calculation:
Each of the first nine digits of the 10-digit ISBN—excluding the check digit itself—is multiplied by its (integer) weight, descending from 10 to 2, and the sum of these nine products found. The value of the check digit is simply the one number between 0 and 10 which, when added to this sum, means the total is a multiple of 11.
So, for the **ISBN 87–11–07559–7**, we would calculate:

```
8×10 + 7×9 + 1×8 + 1×7 + 0×6 + 7×5 + 5×4 + 5×3 + 9×2 =
    80 + 63 + 8 + 7 + 0 + 35 + 20 + 15 + 18 = 246.
```

Now 246 is between 242 = 22×11 and 253 = 23×11. We need to add 7 to 246 in order to get 253, so 7 is the check digit.

The calculation for the **ISBN 0–471–19047–0**:

```
0×10 + 4×9 + 7×8 + 1×7 + 1×6 + 9×5 + 0×4 + 4×3 + 7×2 =
     0 + 36 + 56 + 7 + 6 + 45 + 0 + 12 + 14 = 176.
```

As 176 = 16×11, it is a multiple of 11 so the check digit is 0.

When the check digit turns out to be "10" an "X" is written in the ISBN check digit place ["X" is the Roman numeral for 10] so that there are exactly 10 symbols used in an ISBN.

ISBN-13 check digit calculation:

A 13-digit number is created by starting the number with (978 or 979) followed by the first 9 digits of the current ISBN and finishing with a new check digit. The new check digit is calculated in a different way than the old 10-digit ISBN check digit that we have been talking about.

The ISBN-13 check digit, which is the last digit of the ISBN, must range from 0 to 9 and must be such that the sum of all the twelve digits, each multiplied by its (integer) weight, alternating between 1 and 3, is a multiple of 10.

For ISBN **978-92-95055-02-5.**
9×1 + 7×3 + 8×1 + 9×3 + 2×1 + 9×3 + 5×1 + 0×3 + 5×1 + 5×3 + 0×1 + 2×3 = 125.
125%10 = 5
10 – 5 = 5 → our ISBN number is, indeed, legitimate.

For ISBN **978-0-306-40615-?.**
9×1 + 7×3 + 8×1 + 0×3 + 3×1 + 0×3 + 6×1 + 4×3 + 0×1 + 6×3 + 1×1 + 5×3 = 93.
93%10 = 3
10 – 3 = 7 → our ISBN number check digit is 7.

For ISBN **978-0-306-48615-7. Mistakenly 9th digit read as 8 instead of 0.**
9×1 + 7×3 + 8×1 + 0×3 + 3×1 + 0×3 + 6×1 + 4×3 + 8×1 + 6×3 + 1×1 + 5×3 = 101.
101%10 = 1
10 – 1 = 9 → Error detected: as per **978-0-306-48615**, check digit should be 9 but its 7.

For ISBN **978-0-360-40615-7. Mistakenly 6th and 7th digits are swapped.**
9×1 + 7×3 + 8×1 + 0×3 + 3×1 + 6×3 + 0×1 + 4×3 + 0×1 + 6×3 + 1×1 + 5×3 = 105.
105%10 = 5
10 – 5 = 5 → Error detected

The Conversion Of ISBN-10 to ISBN-13

Here's what we're going to do in the seven steps below:

- take the 10 digit ISBN (10 digits)
- drop the last character (9 digits)
- append "978" to the front (12 digits)
- calculate a new check digit and append it on to the end (13 digits)

Our today's coding job is to design three classes:

- ISBN10: stores the information related to ISBN-10
- ISBN13: stores the information related to ISBN-13
- Book: stores information related to book along with its related ISBN.

C-1: Implement the class *ISBN10* to store 10-digit ISBN as given below.

```
class ISBN10
{
     //Data member listed below are already explained above.
    String group;
    String publisher;
    String title;
    String checkDigit;
public:
    ISBN10(String isbn);                                                                   (2)
     //Received the isbn-10 consisting of four parts separated by '-' or ' ' and initialize them with
     related data members.//You don’t need to validate the check digit in the received string.
    void updateISBN10(String isbn); //setter for the ISBN-10                               (4)
    String getGroup(); //getter for the group data member
    String getPublisher(); //getter for the publisher data member
    String getTitle(); //getter for the Title data member
    String getCheckDigit(); //getter for the check digit data member
    String getISBN10InString();                                                            (3)
     //return ISBN-10 in string form which is stored in four data members. The return string will have four
     parts separated by '-' or ' '.
    bool isValidISBN10();                                                                  (1)
     //returns true if the stored isbn is valid considering the check digit it has otherwise false.
    static bool validateISBN10(String s);                                                  (1)
     //receives the ISBN-10 in string format as described earlier and return true if valid ISBN considering
     the check digit otherwise false.
    static int getValidCheckDigit(String isbn);                                            (6)
     //It just checks the first 9 digits of the isbn to know the actual check digit and returns it.
    ISBN13 toISBN13();                                                                     (4)
     //It converts the stored ISBN10 number into ISBN13 and returns it.
};
```

C-2: Implement the class *ISBN13* to store 13-digit ISBN as given below.

```
class ISBN13
{
     //Data member listed below are already explained above.
    String prefix;
    ISBN10 oldISBN;
public:
    ISBN13(String isbn);                                                                    (4)
     //Received the isbn-13 consisting of five parts separated by '-' or ' ' and initialize them with
     related data members.
    void updateISBN13(String isbn); //setter for the ISBN-13                               (4)
    String getPrefix(); //getter for the prefix data member
    String getGroup();
    String getPublisher();
    String getTitle();
    String getCheckDigit();
    String getISBN13InString();                                                            (2)
     //return ISBN-13 in string form. The return string will have five parts separated by '-' or ' '.
    bool isValidISBN13();                                                                  (1)
     //returns true if the stored isbn is valid considering the check digit it has otherwise false.
    static bool validateISBN13(String s);                                                  (1)
     //receives the ISBN-13 in string format as described earlier and return true if valid ISBN considering
     the check digit otherwise false.
    static int getValidCheckDigit(String isbn);                                            (5)
     //It just checks the first 12 digits of the isbn to know the actual check digit and returns it.
};
```

C-3: Implement the class *Book*.

```
class Book                                                                                 (5)
{
    String bookTitle; //Stores book title.
    String author; //Stores author name.
    double price; //Stores book price.
    ISBN13 isbn; //Stores book ISBN-13.
public:
     //Typical constructor and getter/setter for the Book class.
    Book(String bTitle, String auth, double pr, ISBN13 num);
    void setBookTitle(String title);
    void setAuthor(String auth);
    void setPrice(double pr);
    void setISBN13(ISBN13 num);
    String getBookTitle();
    String getAuthor();
    double getPrice();
    void printISBN13();
      //Prints the book ISBN-13 in string form consisting of five parts separated by '-' or ' '.
    void printISBN10();
      //Prints the equivalent ISBN-10 of the stored ISBN-13 in string form consisting of four parts
      separated by '-' or ' '.
};
```

Assumptions:

- It is assumed that the ISBN-10 number will always be of 10 digits with four parts separated by '-' or ' '. Same goes for ISBN-13 but with 13 digits and 5 parts.

Sample Runs

SR#	Sample Run	Console Output

1	ISBN10 num("0-471-19047-0");
	num.getCheckDigit().display(); cout<<'\n';
		0

	num.getPublisher().display(); cout<<'\n';	471

	String s = num.getISBN10InString();
	s.display(); cout<<'\n';
		0-471-19047-0

	num.updateISBN10("87-11-07559-7");
	num.getISBN10InString().display (); cout<<'\n';
		87-11-07559-7

	num.updateISBN10("87-11-07559-5");//check digit is wrong
	cout<<num.isValidISBN10()<<'\n';
		0

	num.updateISBN10("87-11-05759-7");//check digit fine but 5 & 7 swapped
	cout<<num.isValidISBN10()<<'\n';
		0

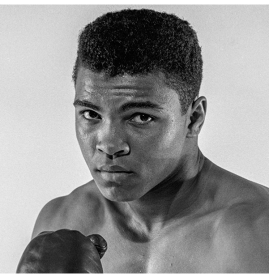
	num.updateISBN10("87-11-07559-7");
	cout<<num.isValidISBN10()<<'\n';
		1

2	ISBN13 num13 ("978-0-306-40615-4");
	cout << ISBN13::getValidCheckDigit("978-0-306-40615-4") << '\n';
		7

	num13.updateISBN13("978-0-306-40615-7");
	cout << num13.isValidISBN13();
		1

3	ISBN10 num10 ("2-1234-5680-2");
	ISBN13 num13(num10.toISBN13());
	String s = num13.getISBN13InString();	

	cout<<"ISBN-13: "; s.display();	ISBN-13: 978-2-1234-5680-3
4	Book b("Izzy Wizzy Let's Get Busy", "XYZ", 45.67, ISBN13("978-2-1234-5680-3"));	
	b.printISBN10(); cout<<'\n';	ISBN10: 2-1234-5680-2
	b.printISBN13();	ISBN13: 978-2-1234-5680-3



"It isn't the mountains ahead to climb that wear you out;
it's the pebble in your shoe".

-- Muhammad Ali –

The champ is referring to the everyday distractions that take us off course and keep us from seeing the bigger picture of our lives.
While the mountain, our life's purpose, is always ahead of us, the pebbles often trip us up. We may perpetuate unhealthy behaviors, patterns and thoughts in our lives by worrying about all the what ifs, could have beens and never-going-to-be's.
Keep your eyes fixed on that mountain! Eliminate the pebbles, one by one.