

Full scores is 100 points. Total scores is 120 points. Anyone who earn 100 points or more will get full scores.

1. This exam is an **open-book and internet-accessible exam**.
 - 1.1. You are **not allowed to use any electronic devices** other than your own storage devices and the computers provided for you in the exam room.
 - 1.2. You are **not allowed to use any earphones, headphones, sound devices, or personal hotspots**.
 - 1.3. You are **not allowed to communicate with anyone** directly or indirectly during the exam except the proctors.
 - 1.4. You are **not allowed to use any forms of AI tools** during the exam.
You must make sure that the IDE that you use does not have any AI tools attached.
 - 1.5. You must **video-record in full screen mode of your display** at all time during the exam
You are not allowed to view anyone's video clips including your own clip.
2. You must **follow the following instructions** to earn full credits on the questions.
 - 2.1. Submit **only the required java source code files**; and no other files.
 - 2.2. **Follow java name convention** for all names, e.g., package names, class names, variable names, constant names, parameter names. **Use only A-Z, a-z, 0-9** for all filenames, folder names (i.e., no non-English characters, no special characters).
 - 2.3. **Format your code** perfectly, i.e., use a proper indentation. **Do not use any comment in your code where they are not real comments**.
 - 2.4. **Do only what the instructions ask you to do**. Do not print anything out in the methods that are not supposed to print anything out. Print text out only in the methods that test the correctness of other methods. Put "main" method only in the main class for testing the correctness of other classes.
 - 2.5. **Do not add any fields, methods, or classes that are not mentioned in the instructions**.
3. Use **the first two digits of your student id** to replace XX and **the last three digits of your student id** to replace YYY in the following instructions.
4. Put the following comment lines above every class in every files you submit.

```
(MainXXXXY.java, UXXXXY.java, BookXXXXY.java, BookListXXXXY.java)
//StudentID: YOUR_11_DIGIT_STUDENT_ID
//StudentName: YOUR_FIRSTNAME_AND_LASTNAME
e.g.,
//StudentID: 60130500999
//StudentName: Kriengkrai Porkaew
```

5. (20 points) Create a **public** class named "**MainXXXXY**" in "**testXXXXY**" package.

Create a **public static void main(String[] args)** method that calls the following

static methods in this class to test all other classes and methods. You must **import** related classes properly.

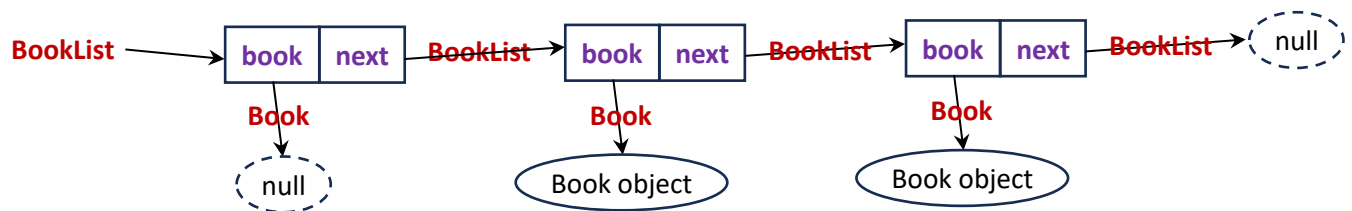
- 5.1. "**testUniqueYYY**" method to test the correctness of "**uniqueYYY**" method in "**UXXXXY**" class in "**utilXXXXY**" package in Question 6.3 by printing the result out.
- 5.2. "**testCheckDigitYYY**" method to test the correctness of "**checkDigitYYY**" method in "**UXXXXY**" class in "**utilXXXXY**" package in Question 6.4 by printing the result out.
- 5.3. "**testBookYYY**" method to test the correctness of "**BookXXXXY**" class in "**libYYY**" package in Question 7 by testing every methods in the class.
- 5.4. "**testBookListYYY**" method to test the correctness of "**BookListXXXXY**" class in "**libYYY**" package in Question 8 by testing every methods in the class including the one you add.

6. (30 points) Create a **public final** class named **"UXXYYY"** in **"utilXXYYY"** package.
- 6.1. Create a **private static** field of type **"long"** named **"seedYYY"** and set it to your full 11-digit student id.
 - 6.2. Create a **private** empty constructor that does nothing.
 - 6.3. Create a **public static** **"uniqueYYY"** method that does not receive any parameters and returns a **"long"**.
This method uses the **Math.random** method to uniformly random an **"int"** between **1** and **9** including **1** and **9**, and uses this random **int** to increment the value of **"seedYYY"**.
This method returns the old value of **"seedYYY"** that has not been incremented.
 - 6.4. Create a **public static** **"checkDigitYYY"** method that receives a parameter of type **"long"** named **"number"** and returns an **"int"**. This method returns **-1** if the input is a negative number. Otherwise, this method subtracts the summation of all odd-position digits with the summation of all even-position digits. (Count the rightmost position of a number as position 1). If the result of this subtraction is a negative number, multiply it by **-1**. This method returns the remainder of the result divided by **10**.
You must not convert any number to a **String**.
You must use some form of loops (e.g., a for loop, a while loop, or a do-while loop) to solve this problem.
Example: if the input is **926184**,
the summation of all odd-position digits will be **4+1+2 = 7** and
the summation of all even-position digits will be **8+6+9 = 23**.
The result of the subtraction of the two numbers will be **7 - 23 = -16**.
Because this number is a negative number, this method will multiply it by **-1** and the result will become **16**.
This method will divide the result by **10** and return the remainder which is **6**.
7. (50 points) Create a **public** class named **"BookXXYYY"** in **"libYYY"** package.
- 7.1. Create **three private final** fields named **"isbnYYY"** (**long**), **"titleYYY"** (**String**), and **"priceYYY"** (**double**).
 - 7.2. Create a **public** constructor that receives two parameters named **"titleYYY"** (**String**) and **"priceYYY"** (**double**). This method throws an **IllegalArgumentException** with an **"Invalid Initialization"** message if the **"titleYYY"** parameter is **null** or is a blank string or the **"priceYYY"** parameter is zero or less. Otherwise, this method will use the two parameters to set the related two fields and generate the **"isbnYYY"** field. The **"isbnYYY"** field is generated as follows:
 - 7.2.1. Call the **"uniqueYYY"** method in Question 6.3 and store the result in **"temp"** variable.
 - 7.2.2. Send **"temp"** to the **"checkDigitYYY"** method in Question 6.4 and store the result in **"digit"** variable.
 - 7.2.3. Set the value of **"isbnYYY"** field to be **"temp" * 10 + "digit"**.
 - 7.3. Create all three getters; one for each field.
 - 7.4. Create an **@Override toString** method that returns a **string** in the following format:
"Book (nn-nnnnn-nnnn-n, titleYYY, priceYYY)" where **"titleYYY"** is the value of the **"titleYYY"** field, **"priceYYY"** is the value of the **"priceYYY"** field with 2 decimal points (i.e., **"%.2f"**), and **"nn-nnnnn-nnnn-n"** is the value of the **"isbnYYY"** field with leading-zero-and-hyphen format (i.e., **"%02d-%05d-%04d-%1d"**). You may assume that the **"isbnYYY"** field is always a 12-digit number.
Example: if the **"isbnYYY"** is **601305009998**, the format will be **"60-13050-0999-8"**.
You may use **String.format("%02d-%05d-%04d-%1d", 60, 13050, 999, 8)** method to do so.
Note: you can get the last 4-digits of a number by calculating the remainder of that number divided by **10000**.
Example: if the number is **20231101**, the last 4-digit is **1101** which is **= 20231101 % 10000**.

- 7.5. Create an **@Override hashCode** method that returns the value of "isbnYYY" as an **int**.
- 7.6. Create an **@Override equals** method that returns **true** only if this object is exactly the same as its parameters (i.e., same reference). Otherwise, it returns **false**.
8. (20 points) Given the following class.

```
package libYYY;

//StudentID: ????????????
//StudentName: ???????????? ????????????
public class BookListXXYYY {
    private final BookXXYYY book;
    private BookListXXYYY next;
    private BookListXXYYY (BookXXYYY book) { this.book = book; }
    public static BookListXXYYY newList() { return new BookListXXYYY(null); }
    public boolean add(BookXXYYY book) {
        if (book == null) return false;
        var current = this;
        while (current.next != null) {
            current = current.next;
            if (current.book.equals(book)) return false;
        }
        current.next = new BookListXXYYY(book);
        return true;
    }
    @Override
    public String toString() {
        var current = this;
        var sb = new StringBuilder();
        sb.append("BookList{");
        while ((current=current.next) != null) {
            sb.append("\n ").append(current.book);
        }
        return sb.append("}").toString();
    }
}
```



- 8.1. Add to this class a **public "totalPriceYYY"** method that returns a **double** but does not receive any parameters. This method returns the summation of the prices of all "BookXXYYY" in the "BookListXXYYY" except the first "BookXXYYY" in the list because it is null.
- 8.2. Add to this class a **public "removeYYY"** method that receives a parameter named "book" of type "BookXXYYY". This method returns **false** if the "book" parameter is not in the list. Otherwise, it removes the "book" from the list and returns **true**.

```

SOLUTION =====
//PATH test60999/Main60999.java -----
package test60999;

import lib999.Book60999;
import lib999.BookList60999;
import util60999.U60999;

public class Main60999 { /* 5.(20 points) */
    public static void main(String[] args) {
        testUnique999(); //5.1
        testCheckDigit999(); //5.2
        testBook999(); //5.3
        testBookList999(); //5.4
    }

    static void testUnique999() { /* 5.1 */
        System.out.println("### Test U60999.unique999 method ###");
        System.out.println(U60999.unique999());
        System.out.println(U60999.unique999());
        System.out.println(U60999.unique999());
    }

    static void testCheckDigit999() { /* 5.2 */
        System.out.println("### Test U60999.checkDigit999 method ###");
        long num;
        num = 60130500999L;
        System.out.format("check digit of %d = %d\n", num, U60999.checkDigit999(num));
        num = -123456;
        System.out.format("check digit of %d = %d\n", num, U60999.checkDigit999(num));
        num = 109030L;
        System.out.format("check digit of %d = %d\n", num, U60999.checkDigit999(num));
        num = 0L;
        System.out.format("check digit of %d = %d\n", num, U60999.checkDigit999(num));
    }

    static void testBook999() { /* 5.3 */
        System.out.println("### Test Book60999 class ###");
        Book60999 b1, b2, b3;
        b1 = new Book60999("First One", 123.45); // constructor
        b2 = new Book60999("First One", 123.45);
        b3 = b1;
        System.out.println("b1: " + b1); // toString
        System.out.println("b2: " + b2);
        System.out.println("b3: " + b3);
        System.out.println("b1.getIsbn999: " + b1.getIsbn999()); // getters
        System.out.println("b1.getTitle999: " + b1.getTitle999());
        System.out.println("b1.getPrice999: " + b1.getPrice999());
        System.out.println("b1.hashCode: " + b1.hashCode()); // hashCode
        System.out.println("b1==b2 ? " + (b1==b2));
        System.out.println("b1.equals(b2) ? " + (b1.equals(b2))); // equals
        System.out.println("b1==b3 ? " + (b1==b3));
        System.out.println("b1.equals(b3) ? " + (b1.equals(b3)));
        try {
            System.out.println("Try: " + new Book60999(null, 1.0)); // constructor
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
        try {
            System.out.println("Try: " + new Book60999(" ", 1.0));
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
        try {
            System.out.println("Try: " + new Book60999("ok", 0.0));
        } catch (Exception e) {
            System.out.println("Error: " + e);
        }
    }
}

```

```

static void testBookList999() { /* 5.4 */
    System.out.println("### Test BookList60999 class ###");

    BookList60999 books = BookList60999.newList(); // newList
    Book60999 b0, b1, b2;

    System.out.format("BookList books: %s\n", books); // toString
    System.out.println("BookList books.totalPrice(): " + books.totalPrice999()); // totalPrice

    b0 = new Book60999("BookOne",100.50);
    System.out.format("books.add(%s): %b\n", b0, books.add(b0)); // add
    System.out.format("BookList books: %s\n", books);
    System.out.println("BookList books.totalPrice(): " + books.totalPrice999());

    b1 = new Book60999("BookTwo",240.25);
    System.out.format("books.add(%s): %b\n", b1, books.add(b1));
    System.out.format("BookList books: %s\n", books);
    System.out.println("BookList books.totalPrice(): " + books.totalPrice999());

    b2 = new Book60999("BookThree",1005.00);
    System.out.format("books.add(%s): %b\n", b2, books.add(b2));
    System.out.format("BookList books: %s\n", books);
    System.out.println("BookList books.totalPrice(): " + books.totalPrice999());

    System.out.format("books.add(%s) {duplicate}: %b\n", b1, books.add(b1));
    System.out.format("BookList books: %s\n", books);
    System.out.println("BookList books.totalPrice(): " + books.totalPrice999());

    System.out.format("books.remove999(%s): %b\n", b1, books.remove999(b1)); // remove
    System.out.format("BookList books: %s\n", books);
    System.out.println("BookList books.totalPrice(): " + books.totalPrice999());

    System.out.format("books.remove999(%s) (not found): %b\n", b1, books.remove999(b1));
    System.out.format("BookList books: %s\n", books);
    System.out.println("BookList books.totalPrice(): " + books.totalPrice999());

    System.out.format("books.remove999(%s): %b\n", b2, books.remove999(b2));
    System.out.format("BookList books: %s\n", books);
    System.out.println("BookList books.totalPrice(): " + books.totalPrice999());

    System.out.format("books.remove999(%s): %b\n", b0, books.remove999(b0));
    System.out.format("BookList books: %s\n", books);
    System.out.println("BookList books.totalPrice(): " + books.totalPrice999());
}
}

//PATH util60999/U60999.java -----
package util60999;

public class U60999 { /* 6.(30 points) */
    private static long seed999 = 60130500999L; /* 6.1 */
    private U60999() { } /* 6.2 */
    public static long unique999() { /* 6.3 */
        long result = seed999;
        seed999 += 1 + (int) (9 * Math.random());
        return result;
    }
    public static int checkDigit999(long number) { /* 6.4 */
        if (number < 0) return -1;
        int sum = 0;
        for (int dir = 1; number > 0; dir *= -1) {
            sum += dir * (int) (number % 10);
            number /= 10;
        }
        if (sum<0) sum *= -1;
        return sum % 10;
    }
}

```

```
//PATH lib999/Book60999.java -----
package lib999;
```

```
import util60999.U60999;
```

```
public class Book60999 { /* 7.(50 points) */
```

```
    private final long isbn999; /* 7.1 */
    private final String title999;
    private final double price999;
```

```
    public Book60999(String title999, double price999) { /* 7.2 */
        if (title999 == null || title999.isBlank() || price999 <= 0.0) {
            throw new IllegalArgumentException("Invalid Initialization");
        }
        this.title999 = title999;
        this.price999 = price999;
        long temp = U60999.unique999();
        this.isbn999 = temp * 10 + U60999.checkDigit999(temp);
    }
```

```
    public long getIsbn999() { return isbn999; } /* 7.3 */
    public String getTitle999() { return title999; }
    public double getPrice999() { return price999; }
```

```
@Override
```

```
public String toString() { /* 7.4 */
    return String.format("Book(%02d-%05d-%04d-%1d,%s,%.2f)"
        , isbn999 / 1_00000_0000_0L
        , isbn999 / 1_0000_0 % 1_00000
        , isbn999 / 1_0 % 1_0000
        , isbn999 % 1_0
        , title999, price999);
}
```

```
@Override
```

```
public int hashCode() { return (int) isbn999; } /* 7.5 */
```

```
@Override
```

```
public boolean equals(Object obj) { return this == obj; } /* 7.6 */
```

```
}
```

```
//PATH lib999/BookList60999.java -----
package lib999;
```

```
public class BookList60999 { /* 8.(20 points) */
```

```
    private final Book60999 book;
    private BookList60999 next;
    private BookList60999 (Book60999 book) { this.book = book; }
    public static BookList60999 newList() { return new BookList60999(null); }
    public boolean add(Book60999 book) {
        if (book == null) return false;
        var current = this;
        while (current.next != null) {
            current = current.next;
            if (current.book.equals(book)) return false;
        }
        current.next = new BookList60999(book);
        return true;
    }
```

```
@Override
```

```
public String toString() {
    var current = this;
    var sb = new StringBuilder();
    sb.append("BookList{");
    while ((current=current.next) != null) {
        sb.append("\n ").append(current.book);
    }
    return sb.append("}").toString();
}
```

```
public double totalPrice999() { /* 8.1 */
    double sum = 0.0;
    var current = this;
    while (current.next != null) {
        current = current.next;
        sum += current.book.getPrice999();
    }
    return sum;
}

public boolean remove999(Book60999 book) { /* 8.2 */
    if (book == null) return false;
    var current = this;
    while (current.next != null) {
        if (current.next.book.equals(book)) {
            current.next = current.next.next;
            return true;
        }
        current = current.next;
    }
    return false;
}
}
```

END OF SOLUTION =====

PROGRAM OUTPUT =====

```
### Test U60999.unique999 method ###
60130500999
60130501001
60130501004
### Test U60999.checkDigit999 method ###
check digit of 60130500999 = 8
check digit of -123456 = -1
check digit of 109030 = 3
check digit of 0 = 0
### Test Book60999 class ###
b1: Book(60-13050-1011-2,First One,123.45)
b2: Book(60-13050-1016-3,First One,123.45)
b3: Book(60-13050-1011-2,First One,123.45)
b1.getIsbn999: 601305010112
b1.getTitle999: First One
b1.getPrice999: 123.45
b1.hashCode: 9588672
b1==b2 ? false
b1.equals(b2) ? false
b1==b3 ? true
b1.equals(b3) ? true
Error: java.lang.IllegalArgumentException: Invalid Initialization
Error: java.lang.IllegalArgumentException: Invalid Initialization
Error: java.lang.IllegalArgumentException: Invalid Initialization
### Test BookList60999 class ###
BookList books: BookList{}
BookList books.totalPrice(): 0.0
books.add(Book(60-13050-1021-3,BookOne,100.50)): true
BookList books: BookList{
    Book(60-13050-1021-3,BookOne,100.50)}
BookList books.totalPrice(): 100.5
books.add(Book(60-13050-1024-0,BookTwo,240.25)): true
BookList books: BookList{
    Book(60-13050-1021-3,BookOne,100.50)
    Book(60-13050-1024-0,BookTwo,240.25)}
BookList books.totalPrice(): 340.75
books.add(Book(60-13050-1032-3,BookThree,1005.00)): true
BookList books: BookList{
    Book(60-13050-1021-3,BookOne,100.50)
    Book(60-13050-1024-0,BookTwo,240.25)
    Book(60-13050-1032-3,BookThree,1005.00)}
BookList books.totalPrice(): 1345.75
books.add(Book(60-13050-1024-0,BookTwo,240.25)) {duplicate}: false
BookList books: BookList{
    Book(60-13050-1021-3,BookOne,100.50)
    Book(60-13050-1024-0,BookTwo,240.25)
    Book(60-13050-1032-3,BookThree,1005.00)}
BookList books.totalPrice(): 1345.75
books.remove999(Book(60-13050-1024-0,BookTwo,240.25)): true
BookList books: BookList{
    Book(60-13050-1021-3,BookOne,100.50)
    Book(60-13050-1032-3,BookThree,1005.00)}
BookList books.totalPrice(): 1105.5
books.remove999(Book(60-13050-1024-0,BookTwo,240.25)) (not found): false
BookList books: BookList{
    Book(60-13050-1021-3,BookOne,100.50)
    Book(60-13050-1032-3,BookThree,1005.00)}
BookList books.totalPrice(): 1105.5
books.remove999(Book(60-13050-1032-3,BookThree,1005.00)): true
BookList books: BookList{
    Book(60-13050-1021-3,BookOne,100.50)}
BookList books.totalPrice(): 100.5
books.remove999(Book(60-13050-1021-3,BookOne,100.50)): true
BookList books: BookList{}
BookList books.totalPrice(): 0.0
```

END OF PROGRAM OUTPUT =====