Appendices

Screenshots for Lab2

Page

Exercise 1 : Learning OOP Structure in Java

Exercise 2 : Fill in the blank to complete an OOP Java application

Exercise 3 : Learning and practicing Interface and Inheritance in Java

Exercise 1: Create an OOP Java Application

Step 1.1 Build an OOP Java Application

You may use copy & paste function to copy below code for two Java Classes in Exercise 1

[Score]

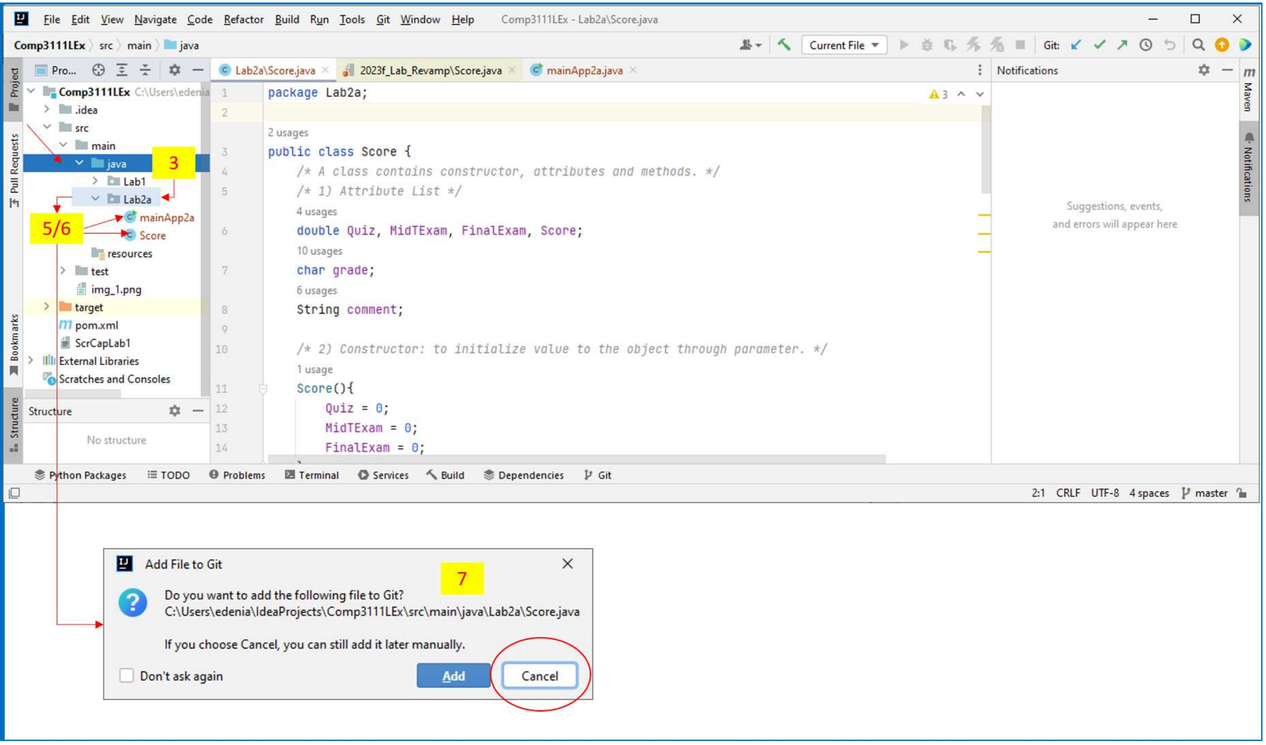
```
package Lab2a;

public class Score {
    /* A class contains constructor[C], attributes[A] and methods[M]. */
    /* [A] Attribute List */
    double Quiz, MidTExam, FinalExam, Score;
    char grade;
    String comment;

    /* [C] Constructor: to initialize value to the object through parameter. */
    Score(){
        Quiz = 0;
        MidTExam = 0;
        FinalExam = 0;
    }

    /* [M] Method: It includes procedure and function. */
    /* Below are Procedures[M1]. Procedure is a sub program to run several process,
     but not return value(s) */
    void setQuiz(double x) {
        Quiz = x ;
    }
    void setMidTExam(double x) {
        MidTExam = x;
    }
    void setFinalExam(double x) {
        FinalExam = x;
    }

    /* Below are Functions[M2]. Function is statement that creates to run
     and return value(s) */
    double getScore() {
        Score = 0.2*Quiz + 0.3*MidTExam + 0.5*FinalExam;
        return Score;
    }
    char getGrade() {
        if(Score >=80 && Score <= 100)
            grade = 'A';
        else if(Score >= 65 && Score < 80)
            grade = 'B';
        else if(Score > 50 && Score < 65)
            grade = 'C';
        else if(Score > 40 && Score < 50)
            grade = 'D';
        else
            grade = 'E';
        return grade;
    }
    String getComment(){
        if(grade == 'A')
            comment = "Very Good";
        else if(grade == 'B')
            comment = "Good";
        else if(grade == 'C')
            comment = "Not Bad";
        else if(grade == 'D')
            comment = "Bad";
        else
            comment = "Very Bad";
        return comment;
    }
}
```

[mainApp2a]

```
package Lab2a;

public class mainApp2a {
    public static void main(String[] args) {
        Score k = new Score();
        k.setQuiz(90);
        k.setMidTExam(70);
        k.setFinalExam(85);
        System.out.println("Quiz        : "+k.Quiz);
        System.out.println("Midterm Exam: "+k.MidTExam);
        System.out.println("Final Exam  : "+k.FinalExam);
        System.out.println("Total Score : "+k.getScore());
        System.out.println("Letter Grade: "+k.getGrade());
        System.out.println("Comment     : "+k.getComment());
    }
}
```

Step 1.1 Build an OOP Java Application (cont …)

LAB-2 | Exercise-1 | Step 1.1



```java
package Lab2a;

2 usages
public class Score {
    /* A class contains constructor, attributes and methods. */
    /* 1) Attribute List */
    4 usages
    double Quiz, MidTExam, FinalExam, Score;
    10 usages
    char grade;
    6 usages
    String comment;

    /* 2) Constructor: to initialize value to the object through parameter. */
    1 usage
    Score(){
        Quiz = 0;
        MidTExam = 0;
        FinalExam = 0;
```

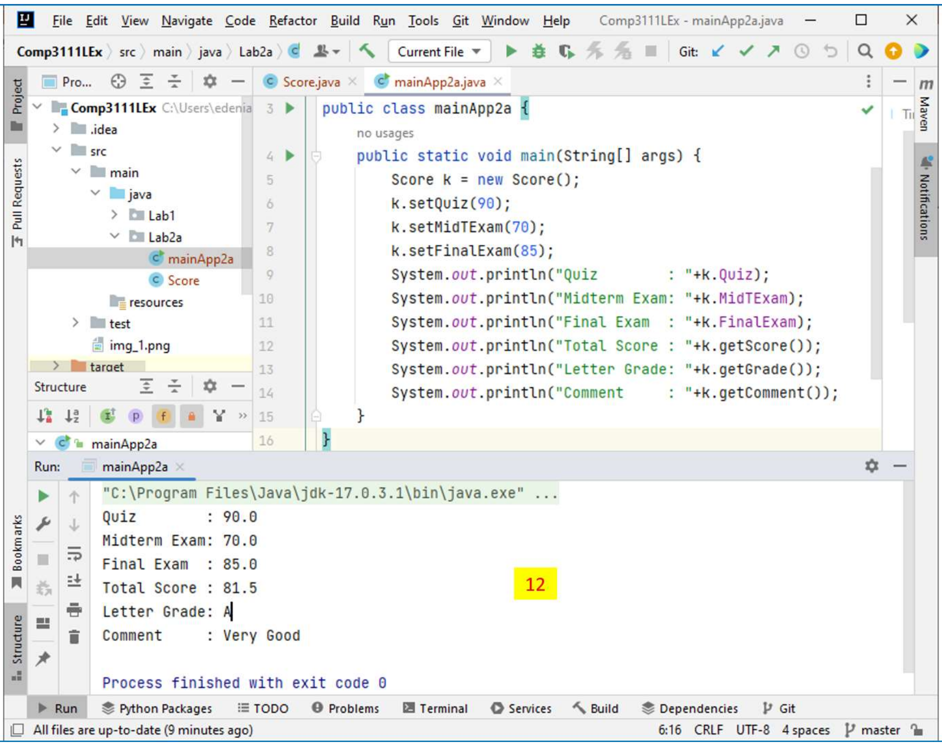
Add File to Git

Do you want to add the following file to Git?
C:\Users\edenia\IdeaProjects\Comp3111LEx\src\main\java\Lab2a\Score.java

If you choose Cancel, you can still add it later manually.

☐ Don't ask again   Add   Cancel

LAB-2 | Exercise-1 | Step 1.1



```java
public class mainApp2a {
    no usages
    public static void main(String[] args) {
        Score k = new Score();
        k.setQuiz(90);
        k.setMidTExam(70);
        k.setFinalExam(85);
        System.out.println("Quiz        : "+k.Quiz);
        System.out.println("Midterm Exam: "+k.MidTExam);
        System.out.println("Final Exam  : "+k.FinalExam);
        System.out.println("Total Score : "+k.getScore());
        System.out.println("Letter Grade: "+k.getGrade());
        System.out.println("Comment     : "+k.getComment());
    }
}
```

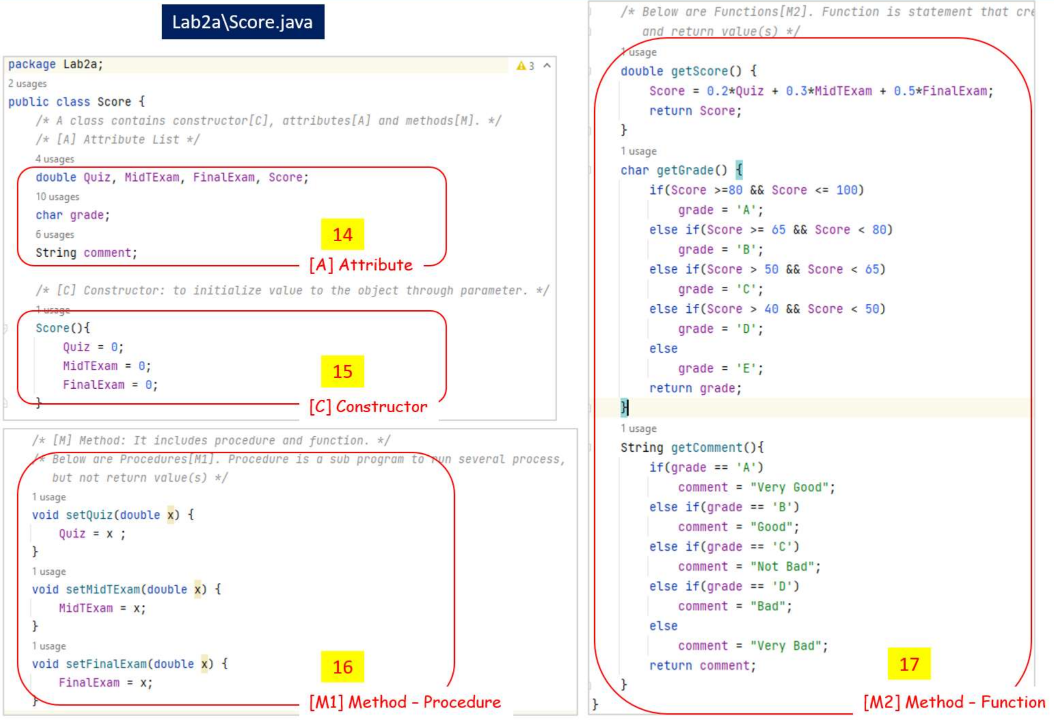
```
"C:\Program Files\Java\jdk-17.0.3.1\bin\java.exe" ...
Quiz        : 90.0
Midterm Exam: 70.0
Final Exam  : 85.0
Total Score : 81.5
Letter Grade: A
Comment     : Very Good

Process finished with exit code 0
```

All files are up-to-date (9 minutes ago)

Step 1.1 Build an OOP Java Application (cont …)

LAB-2 | Exercise-1 | Step 1.1

Lab2a\Score.java



```
package Lab2a;
2 usages
public class Score {
    /* A class contains constructor[C], attributes[A] and methods[M]. */
    /* [A] Attribute List */
    4 usages
    double Quiz, MidTExam, FinalExam, Score;
    10 usages
    char grade;
    6 usages
    String comment;

    /* [C] Constructor: to initialize value to the object through parameter. */
    1 usage
    Score(){
        Quiz = 0;
        MidTExam = 0;
        FinalExam = 0;
    }
```

```
    /* [M] Method: It includes procedure and function. */
    /* Below are Procedures[M1]. Procedure is a sub program to run several process,
       but not return value(s) */
    1 usage
    void setQuiz(double x) {
        Quiz = x ;
    }
    1 usage
    void setMidTExam(double x) {
        MidTExam = x;
    }
    1 usage
    void setFinalExam(double x) {
        FinalExam = x;
    }
```

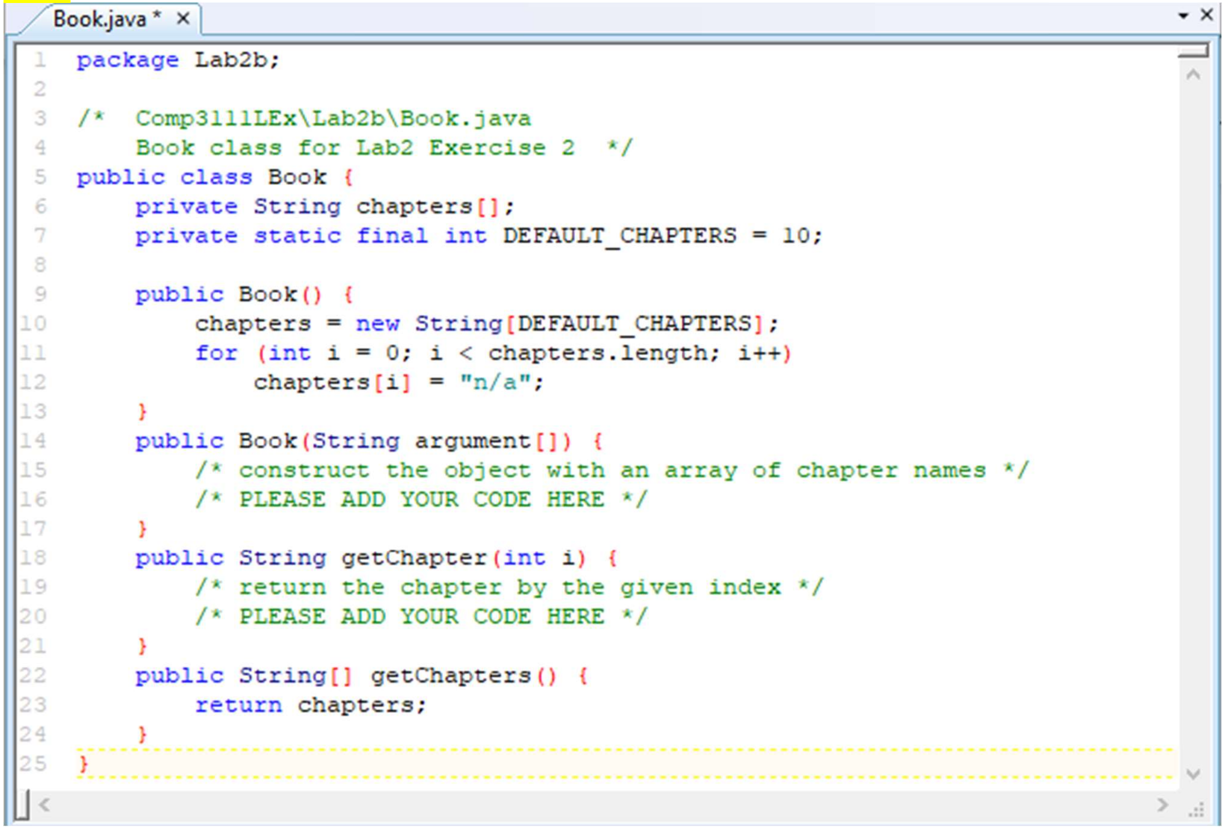
```
    /* Below are Functions[M2]. Function is statement that cre
       and return value(s) */
    1 usage
    double getScore() {
        Score = 0.2*Quiz + 0.3*MidTExam + 0.5*FinalExam;
        return Score;
    }
    1 usage
    char getGrade() {
        if(Score >=80 && Score <= 100)
            grade = 'A';
        else if(Score >= 65 && Score < 80)
            grade = 'B';
        else if(Score > 50 && Score < 65)
            grade = 'C';
        else if(Score > 40 && Score < 50)
            grade = 'D';
        else
            grade = 'E';
        return grade;
    }
    1 usage
    String getComment(){
        if(grade == 'A')
            comment = "Very Good";
        else if(grade == 'B')
            comment = "Good";
        else if(grade == 'C')
            comment = "Not Bad";
        else if(grade == 'D')
            comment = "Bad";
        else
            comment = "Very Bad";
        return comment;
    }
}
```

Exercise 2: Fill in the blank to complete an OOP Java application

Step 2.1 Lab 2 assignment Part-1 – Build OOP Classes

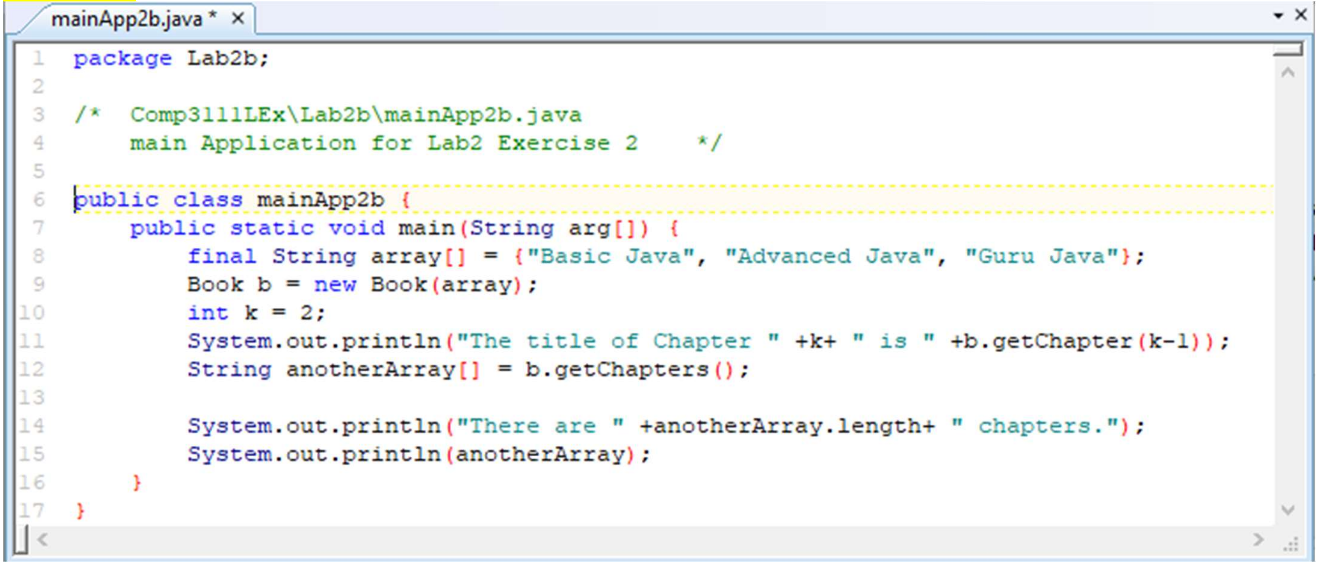
You may copy/type below code for two Java Classes in Exercise 2

[Book]



```java
package Lab2b;

/*  Comp3111LEx\Lab2b\Book.java
    Book class for Lab2 Exercise 2  */
public class Book {
    private String chapters[];
    private static final int DEFAULT_CHAPTERS = 10;

    public Book() {
        chapters = new String[DEFAULT_CHAPTERS];
        for (int i = 0; i < chapters.length; i++)
            chapters[i] = "n/a";
    }
    public Book(String argument[]) {
        /* construct the object with an array of chapter names */
        /* PLEASE ADD YOUR CODE HERE */
    }
    public String getChapter(int i) {
        /* return the chapter by the given index */
        /* PLEASE ADD YOUR CODE HERE */
    }
    public String[] getChapters() {
        return chapters;
    }
}
```

[mainApp2b]

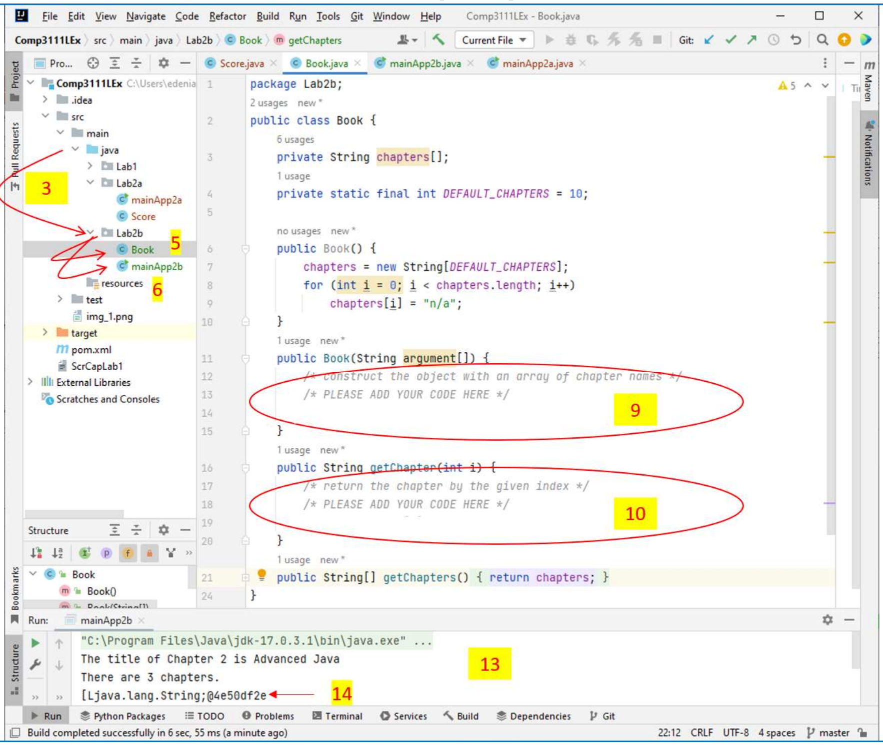


```java
package Lab2b;

/*  Comp3111LEx\Lab2b\mainApp2b.java
    main Application for Lab2 Exercise 2    */

public class mainApp2b {
    public static void main(String arg[]) {
        final String array[] = {"Basic Java", "Advanced Java", "Guru Java"};
        Book b = new Book(array);
        int k = 2;
        System.out.println("The title of Chapter " +k+ " is " +b.getChapter(k-1));
        String anotherArray[] = b.getChapters();

        System.out.println("There are " +anotherArray.length+ " chapters.");
        System.out.println(anotherArray);
    }
}
```

Step 2.1 Lab Assignment Part 1 – Build OOP Classes

** Fill in the blank of code in 2 functions of [Book] Java class **

LAB-2 | Exercise-2 | Step 2.1



```java
package Lab2b;
public class Book {
    private String chapters[];
    private static final int DEFAULT_CHAPTERS = 10;

    public Book() {
        chapters = new String[DEFAULT_CHAPTERS];
        for (int i = 0; i < chapters.length; i++)
            chapters[i] = "n/a";
    }
    public Book(String argument[]) {
        /* construct the object with an array of chapter names */
        /* PLEASE ADD YOUR CODE HERE */

    }
    public String getChapter(int i) {
        /* return the chapter by the given index */
        /* PLEASE ADD YOUR CODE HERE */

    }
    public String[] getChapters() { return chapters; }
}
```

```
"C:\Program Files\Java\jdk-17.0.3.1\bin\java.exe" ...
The title of Chapter 2 is Advanced Java
There are 3 chapters.
[Ljava.lang.String;@4e50df2e
```

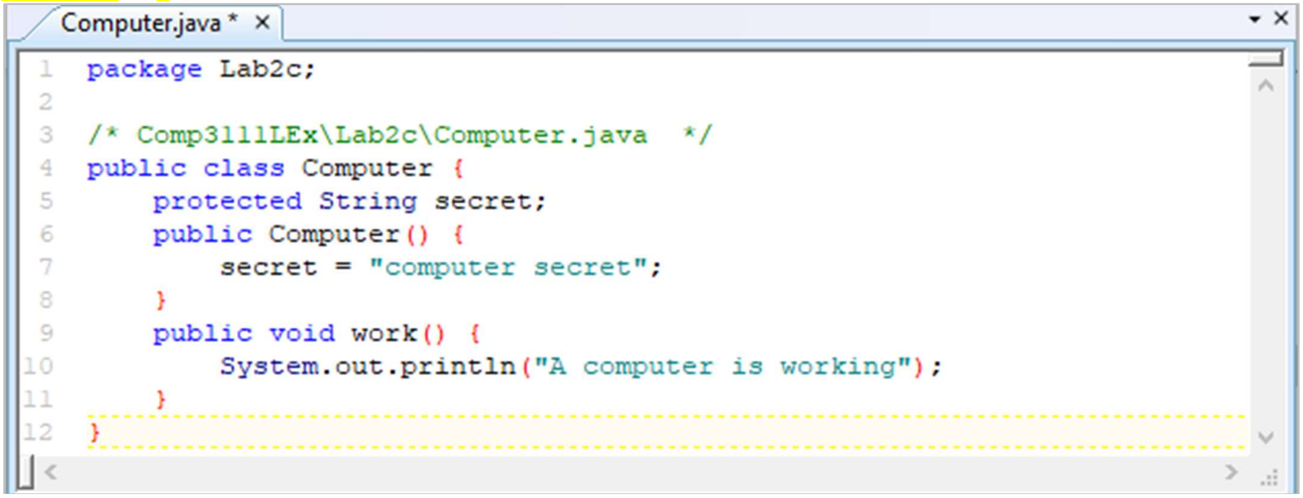
Build completed successfully in 6 sec, 55 ms (a minute ago)

Exercise 3: Learning and practicing Interface and Inheritance in Java

Step 3.1 Create class of Inheritance

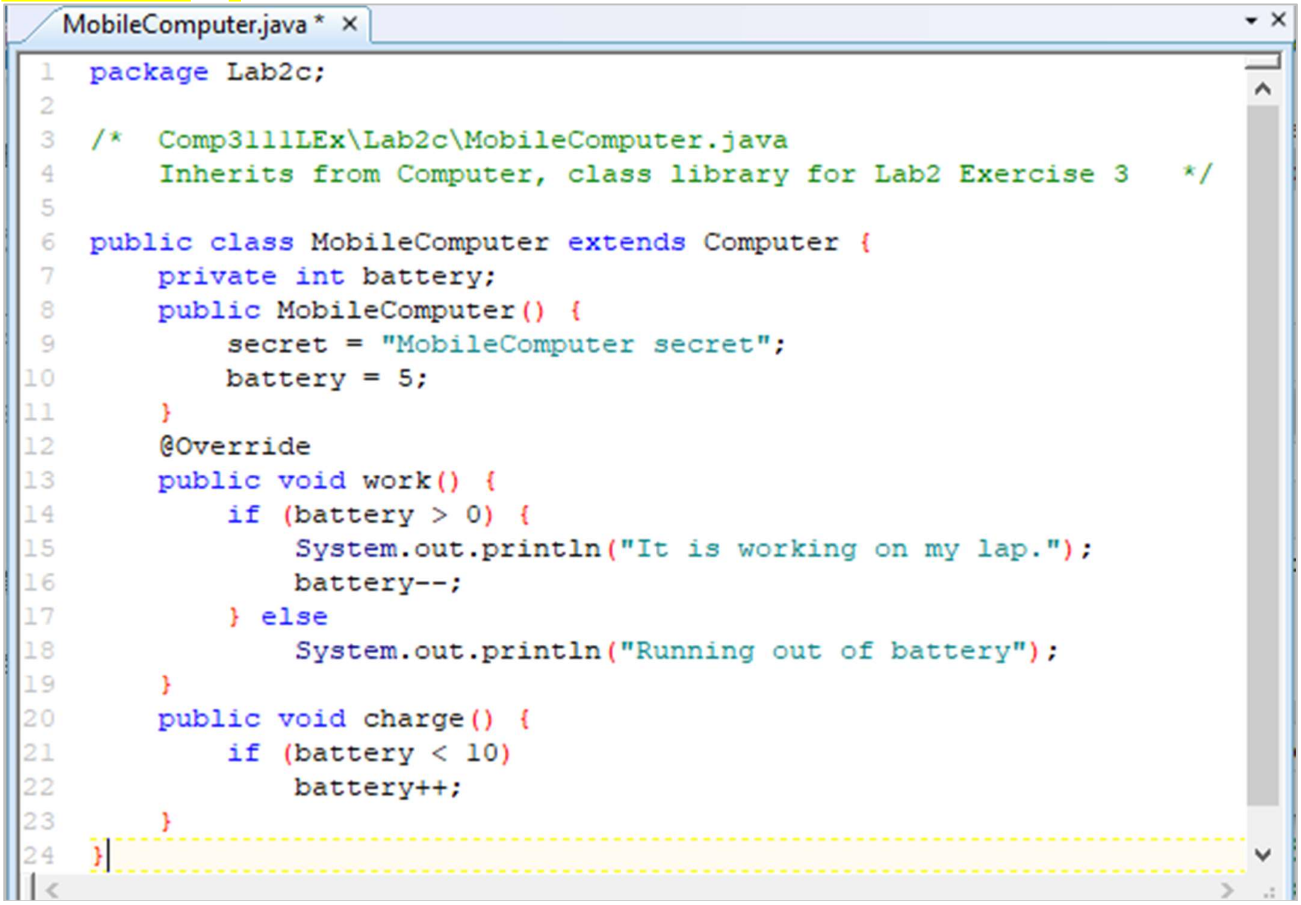
You may copy/type below code for five Java Classes in Exercise 3

[Computer] 8



```java
package Lab2c;

/* Comp3111LEx\Lab2c\Computer.java  */
public class Computer {
    protected String secret;
    public Computer() {
        secret = "computer secret";
    }
    public void work() {
        System.out.println("A computer is working");
    }
}
```

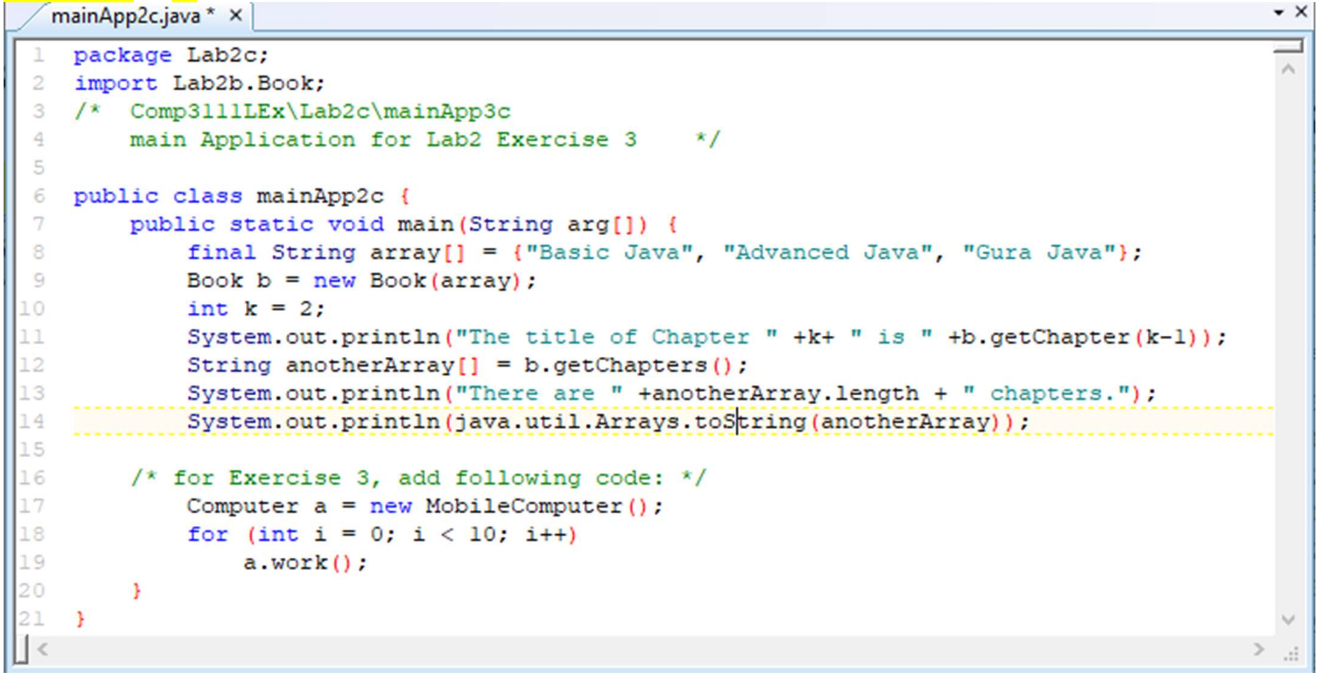
[MobileComputer] 9



```java
package Lab2c;

/*  Comp3111LEx\Lab2c\MobileComputer.java
    Inherits from Computer, class library for Lab2 Exercise 3   */

public class MobileComputer extends Computer {
    private int battery;
    public MobileComputer() {
        secret = "MobileComputer secret";
        battery = 5;
    }
    @Override
    public void work() {
        if (battery > 0) {
            System.out.println("It is working on my lap.");
            battery--;
        } else
            System.out.println("Running out of battery");
    }
    public void charge() {
        if (battery < 10)
            battery++;
    }
}
```

Step 3.1 Create class of Inheritance (cont …)

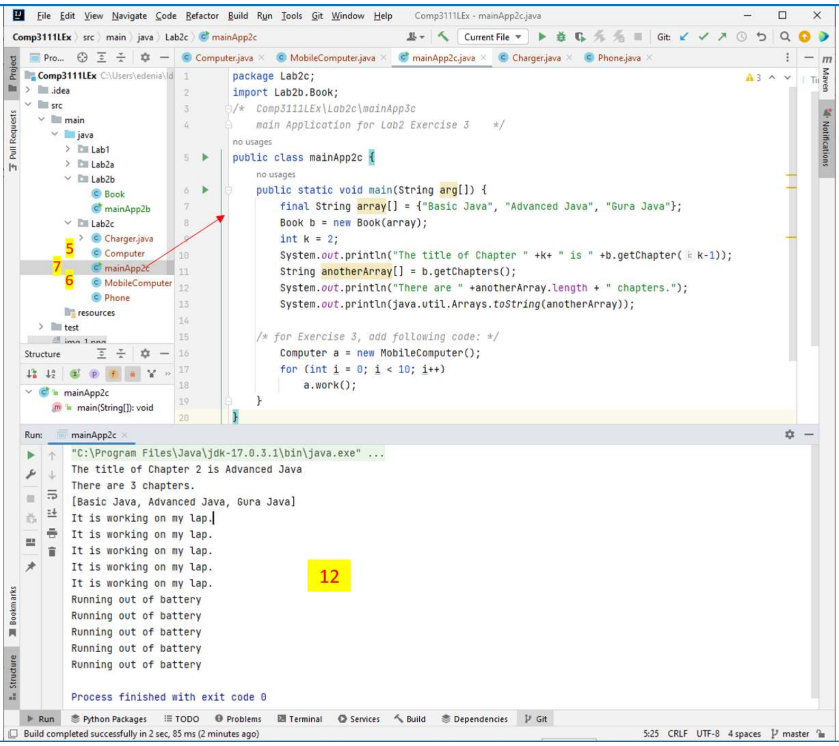
[mainApp2c] 10



```java
package Lab2c;
import Lab2b.Book;
/*  Comp3111LEx\Lab2c\mainApp3c
    main Application for Lab2 Exercise 3    */

public class mainApp2c {
    public static void main(String arg[]) {
        final String array[] = {"Basic Java", "Advanced Java", "Gura Java"};
        Book b = new Book(array);
        int k = 2;
        System.out.println("The title of Chapter " +k+ " is " +b.getChapter(k-1));
        String anotherArray[] = b.getChapters();
        System.out.println("There are " +anotherArray.length + " chapters.");
        System.out.println(java.util.Arrays.toString(anotherArray));

    /* for Exercise 3, add following code: */
        Computer a = new MobileComputer();
        for (int i = 0; i < 10; i++)
            a.work();
    }
}
```

Step 3.1 Create class of Inheritance (cont …)

LAB-2 | Exercise-3 | Step 3.1

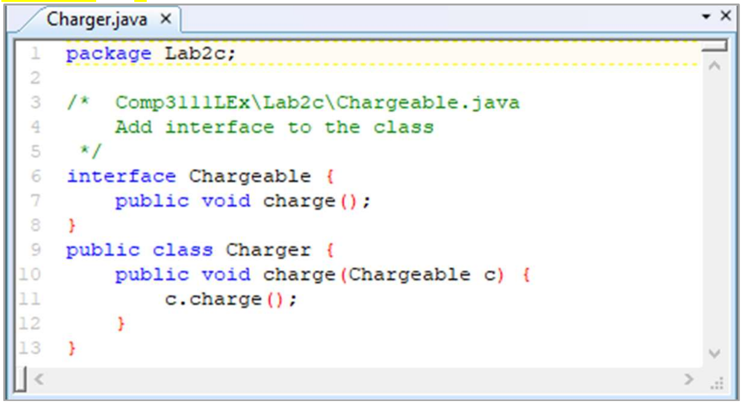


```java
package Lab2c;
import Lab2b.Book;
/*  Comp3111LEx\Lab2c\mainApp3c
    main Application for Lab2 Exercise 3    */
public class mainApp2c {
    public static void main(String arg[]) {
        final String array[] = {"Basic Java", "Advanced Java", "Gura Java"};
        Book b = new Book(array);
        int k = 2;
        System.out.println("The title of Chapter " +k+ " is " +b.getChapter( k-1));
        String anotherArray[] = b.getChapters();
        System.out.println("There are " +anotherArray.length + " chapters.");
        System.out.println(java.util.Arrays.toString(anotherArray));

    /* for Exercise 3, add following code: */
        Computer a = new MobileComputer();
        for (int i = 0; i < 10; i++)
            a.work();
    }
}
```

```
"C:\Program Files\Java\jdk-17.0.3.1\bin\java.exe" ...
The title of Chapter 2 is Advanced Java
There are 3 chapters.
[Basic Java, Advanced Java, Gura Java]
It is working on my lap.
It is working on my lap.
It is working on my lap.
It is working on my lap.
It is working on my lap.
Running out of battery
Running out of battery
Running out of battery
Running out of battery
Running out of battery

Process finished with exit code 0
```

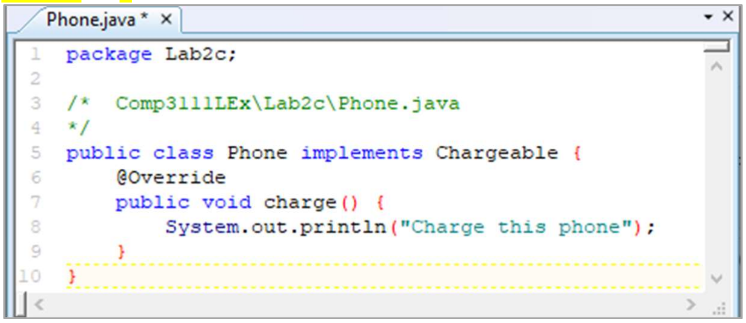
Step 3.2 Add Interface

[Charger] 4



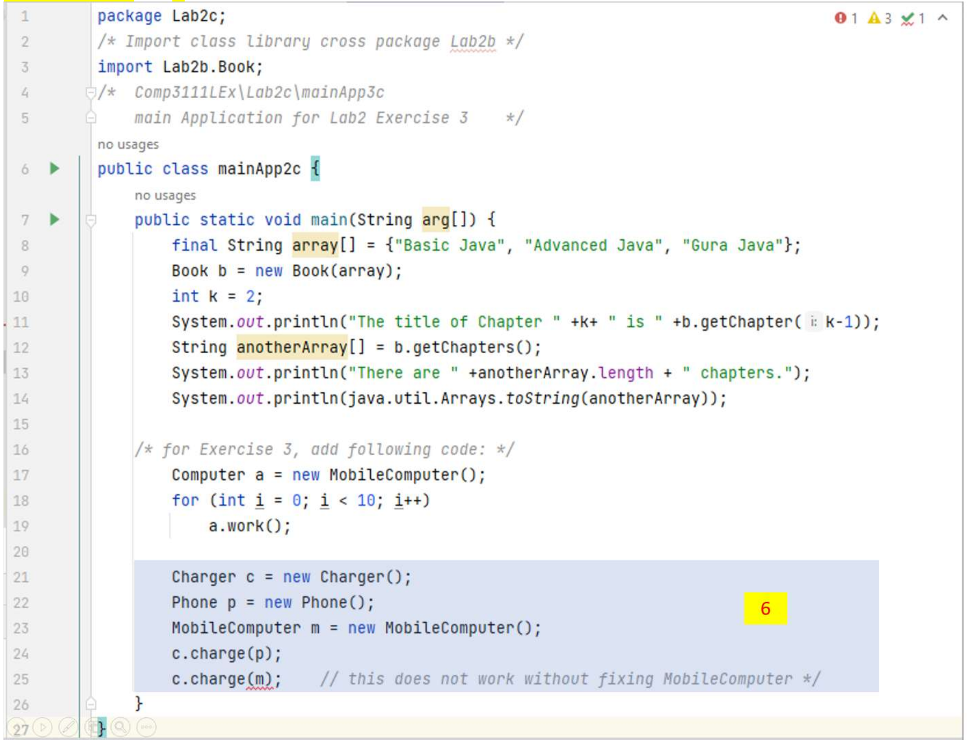
```
package Lab2c;

/*  Comp3111LEx\Lab2c\Chargeable.java
    Add interface to the class
 */
interface Chargeable {
    public void charge();
}
public class Charger {
    public void charge(Chargeable c) {
        c.charge();
    }
}
```

[Phone] 5



```
package Lab2c;

/*  Comp3111LEx\Lab2c\Phone.java
*/
public class Phone implements Chargeable {
    @Override
    public void charge() {
        System.out.println("Charge this phone");
    }
}
```

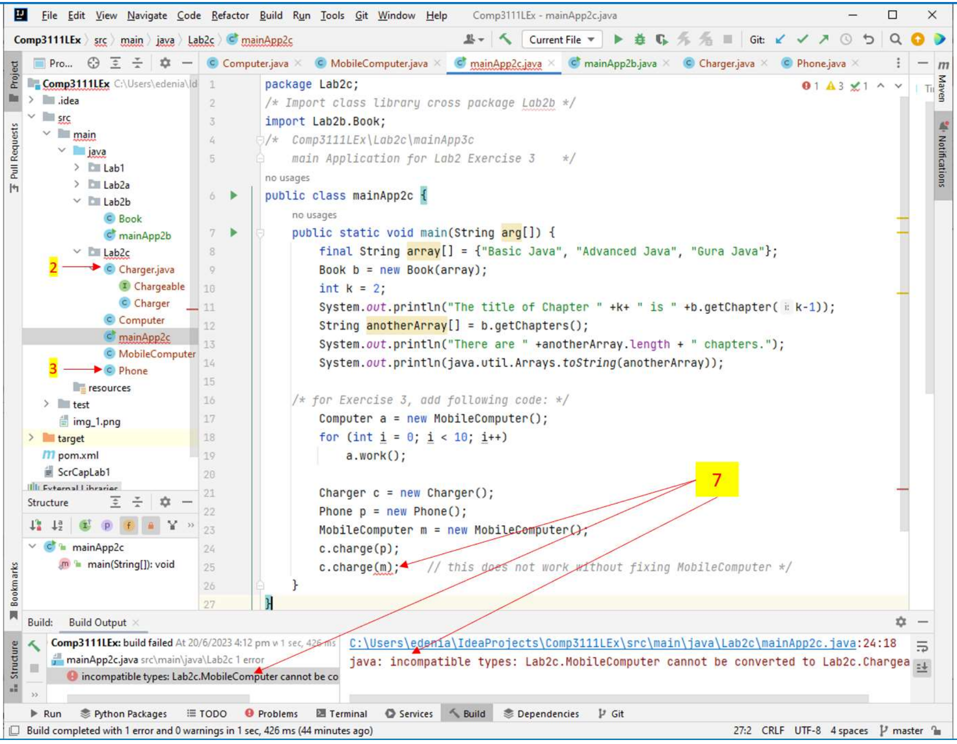
[mainApp2c] 6



```
package Lab2c;
/* Import class library cross package Lab2b */
import Lab2b.Book;
/*  Comp3111LEx\Lab2c\mainApp3c
    main Application for Lab2 Exercise 3    */
no usages
public class mainApp2c {
    no usages
    public static void main(String arg[]) {
        final String array[] = {"Basic Java", "Advanced Java", "Gura Java"};
        Book b = new Book(array);
        int k = 2;
        System.out.println("The title of Chapter " +k+ " is " +b.getChapter( i: k-1));
        String anotherArray[] = b.getChapters();
        System.out.println("There are " +anotherArray.length + " chapters.");
        System.out.println(java.util.Arrays.toString(anotherArray));

    /* for Exercise 3, add following code: */
        Computer a = new MobileComputer();
        for (int i = 0; i < 10; i++)
            a.work();

        Charger c = new Charger();
        Phone p = new Phone();
        MobileComputer m = new MobileComputer();
        c.charge(p);
        c.charge(m);    // this does not work without fixing MobileComputer */
    }
}
```

Step 3.2 Add Interface (cont …)

LAB-2 | Exercise-3 | Step 3.2



```java
package Lab2c;
/* Import class library cross package Lab2b */
import Lab2b.Book;
/*  Comp3111LEx\Lab2c\mainApp3c
    main Application for Lab2 Exercise 3    */
public class mainApp2c {
    public static void main(String arg[]) {
        final String array[] = {"Basic Java", "Advanced Java", "Gura Java"};
        Book b = new Book(array);
        int k = 2;
        System.out.println("The title of Chapter " +k+ " is " +b.getChapter( k-1));
        String anotherArray[] = b.getChapters();
        System.out.println("There are " +anotherArray.length + " chapters.");
        System.out.println(java.util.Arrays.toString(anotherArray));

    /* for Exercise 3, add following code: */
        Computer a = new MobileComputer();
        for (int i = 0; i < 10; i++)
            a.work();

        Charger c = new Charger();
        Phone p = new Phone();
        MobileComputer m = new MobileComputer();
        c.charge(p);
        c.charge(m);    // this does not work without fixing MobileComputer */
    }
}
```

```
Comp3111LEx: build failed At 20/6/2023 4:12 pm w 1 sec, 426 ms
  mainApp2c.java src\main\java\Lab2c 1 error
    incompatible types: Lab2c.MobileComputer cannot be co

C:\Users\edenia\IdeaProjects\Comp3111LEx\src\main\java\Lab2c\mainApp2c.java:24:18
java: incompatible types: Lab2c.MobileComputer cannot be converted to Lab2c.Chargea
```

Build completed with 1 error and 0 warnings in 1 sec, 426 ms (44 minutes ago)