

Lab 1

Getting Started

Objectives

Upon completion of this lab, you should be able to:

- Diagnose simple compilation and runtime errors
- Create a test program for an existing class

Exercise 1: Exploring Java™ Program Errors

In this exercise, you view the source of several simple Java programs, and correct their compilation and runtime errors.

This exercise contains the following sections:

- “Task – Correcting Compilation and Runtime Errors”

Preparation

No preparation is needed for this exercise.



Tool Reference – Tool references used in this exercise:

- Java Development: Java Classes: Opening Java Classes
- Java Development: Java Classes: Modifying Java Classes: Compiling Java Classes
- Java Development: Java Classes: Modifying Java Classes: Executing Java Programs

For this exercise, you work in the `TestProject` project in the `exercises/01_intro/exercisel` directory.

Demonstration – The demonstration for this exercise can be found in the `demos/01_intro/exercisel` directory.



Task – Correcting Compilation and Runtime Errors

In this task, you are presented with four Java programs that contain errors. The errors can be either compilation or runtime errors. Your job is to diagnose and fix those errors so the programs will execute.

Complete the following steps for the files `Test1.java`, `Test2.java`, `Test3.java` and `Test4.java`:



Tool Reference – Java Development: Java Classes: Opening Java Classes

1. Open the Java file.



Tool Reference – Java Development: Java Classes: Modifying Java Classes: Compiling Java Classes

2. Compile the Java file.
3. If a compilation error occurs, identify the source of the error and fix it.
4. Repeat step 2 and step 3 until there are no more errors.



Tool Reference – Java Development: Java Classes: Modifying Java Classes: Executing Java Programs

5. Execute the Java program.
6. If a runtime error occurs, identify the source of the error and fix it.
7. Repeat step 2 and step 3 until there are no more errors.

Exercise 2: Creating a Test Program (Level 1)

In this exercise, you create a test program (also known as a *test harness*) to exercise a pre-provided class. These are the Level 1 instructions that provide additional hints.

Figure 1-1 shows the definition of the `Account` class using a Unified Modeling Language (UML) Class diagram.

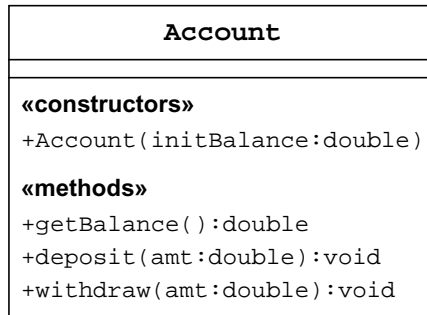


Figure 1-1 Banking Account Class

This exercise contains the following sections:

- “Task 1 – Copying the Account Class”
- “Task 2 – Creating the TestAccount Class”
- “Task 3 – Compiling the TestAccount Class”
- “Task 4 – Running the TestAccount Program”

Preparation

No preparation is needed for this exercise.



Tool Reference – Tool references used in this exercise:

- Java Development: Java Application Projects: Creating Projects
- Java Development: Java Classes: Modifying Java Classes: Copying Java Classes
- Java Development: Java Classes: Creating Java Classes
- Java Development: Java Classes: Modifying Java Classes: Compiling Java Classes
- Java Development: Java Classes: Modifying Java Classes: Executing Java Programs



Demonstration – The demonstration for this exercise can be found in the `demos/01_intro/exercise2` directory.

Task 1 – Copying the Account Class

In this task, you create the `BankPrj` project, and copy the pre-provided `Account` class to the project.



Tool Reference – Java Development: Java Application Projects: Creating Projects

1. Create the `BankPrj` Java Application Project with the following characteristics:

Project Name: **BankPrj**

Project Location: **projects**

Project Folder: **projects/BankPrj**

Set as Main Project: **No**

Create Main Class: **No**



Tool Reference – Java Development: Java Classes: Modifying Java Classes: Copying Java Classes

2. Copy the pre-provided `Account.java` source file from the `resources/01_intro/exercise2` directory to the source package of the `BankPrj` project.

Task 2 – Creating the TestAccount Class

In this task, you complete the following steps to create a new `TestAccount` Java class.



Tool Reference – Java Development: Java Classes: Creating Java Classes

1. Create a new Java class in the `BankPrj` project with the following characteristics:

Class Name: **TestAccount**

Project: **BankPrj**

Location: **Source Packages**

Package: `default package`

2. Edit the source file for the `TestAccount` class to add a `main` method. The `main` method of the `TestAccount` class creates an `Account` object with an initial balance of 100. It deposits 50 to and then withdraws 147 from the `Account` object. Finally, it must print out the balance of the `Account` object to the standard output stream.

Task 3 – Compiling the `TestAccount` Class

Compile the `TestAccount` class, and make necessary changes to correct compilation errors.

Task 4 – Running the `TestAccount` Program

Run the `TestAccount` program. If there are runtime errors, make necessary changes to the `TestAccount` class, recompile it, and run the program again.

The output of the `TestAccount` program should be similar to the following:

```
Final account balance is: 3.0
```

Exercise 2: Creating a Test Program (Level 2)

In this exercise, you create a test harness (a test class) to exercise a pre-provided class. These are the Level 2 instructions that provide additional hints.

This exercise contains the following sections:

- “Task 1 – Copying the Account Class”
- “Task 2 – Creating the TestAccount Class”
- “Task 3 – Compiling the TestAccount Class”
- “Task 4 – Running the TestAccount Program”

Preparation

No preparation is needed for this exercise.



Tool Reference – Tool references used in this exercise:

- Java Development: Java Application Projects: Creating Projects
- Java Development: Java Classes: Modifying Java Classes: Copying Java Classes
- Java Development: Java Classes: Creating Java Classes
- Java Development: Java Classes: Modifying Java Classes: Compiling Java Classes
- Java Development: Java Classes: Modifying Java Classes: Executing Java Programs



Demonstration – The demonstration for this exercise can be found in the `demos/01_intro/exercise2` directory.

Task 1 – Copying the Account Class

In this task, you create the `BankPrj` project, and copy the pre-provided `Account` class to the project.



Tool Reference – Java Development: Java Application Projects: Creating Projects

1. Create the `BankPrj` Java Application Project with the following characteristics:

Project Name: **BankPrj**

Project Location: **projects**

Project Folder: **projects/BankPrj**

Set as Main Project: **No**

Create Main Class: **No**

Tool Reference – Java Development: Java Classes: Modifying Java Classes: Copying Java Classes



2. Copy the pre-provided `Account.java` source file from the `resources/01_intro/exercise2` directory to the source package of the `BankPrj` project.

Task 2 – Creating the TestAccount Class

In this task, you complete the following steps to create a new `TestAccount` Java class.

Tool Reference – Java Development: Java Classes: Creating Java Classes



1. Create a Java class with the following characteristics:

Class Name: **TestAccount**

Project: **BankPrj**

Location: **Source Packages**

Package: **default package**

2. Add the main method.
3. Declare a variable in the main method. The variable has a type of `Account` and a name of `acct`. Initialize the variable by creating an instance of the `Account` class with an initial balance of 100.

4. Use the `deposit` method to add 50 to the account.
5. Use the `withdraw` method to subtract 147 from the account.
6. Use the `getBalance` method to retrieve the new account balance and use the `System.out.println` method to display the balance to the standard output stream.

Task 3 – Compiling the `TestAccount` Class

Compile the `TestAccount` class, and make necessary changes to correct compilation errors.

Task 4 – Running the `TestAccount` Program

Run the `TestAccount` program. If there are runtime errors, make necessary changes to the `TestAccount` class, recompile it and run the program again.

The output of the `TestAccount` program should be similar to the following:

```
Final account balance is: 3.0
```

Exercise 2: Creating a Test Program (Level 3)

In this exercise, you create a test harness (a test class) to exercise a pre-provided class. These are the Level 3 instructions that provide additional hints with code snippets.

This exercise contains the following sections:

- “Task 1 – Copying the Account Class”
- “Task 2 – Creating the TestAccount Class”
- “Task 3 – Compiling the TestAccount Class”
- “Task 4 – Running the TestAccount Program”

Preparation

No preparation is needed for this exercise.

Tool Reference – Tool references used in this exercise:

- Java Development: Java Application Projects: Creating Projects
- Java Development: Java Classes: Modifying Java Classes: Copying Java Classes
- Java Development: Java Classes: Creating Java Classes
- Java Development: Java Classes: Modifying Java Classes: Compiling Java Classes
- Java Development: Java Classes: Modifying Java Classes: Executing Java Programs

Demonstration – The demonstration for this exercise can be found in the `demos/01_intro/exercise2` directory.



Task 1 – Copying the Account Class

In this task, you create the `BankPrj` project, and copy the pre-provided `Account` class to the project.



Tool Reference – Java Development: Java Application Projects: Creating Projects

1. Create the `BankPrj` Java Application Project with the following characteristics:

Project Name: **BankPrj**

Project Location: **projects**

Project Folder: **projects/BankPrj**

Set as Main Project: **No**

Create Main Class: **No**



Tool Reference – Java Development: Java Classes: Modifying Java Classes: Copying Java Classes

2. Copy the pre-provided `Account.java` source file from the `resources/01_intro/exercise2` directory to the source package of the `BankPrj` project.

Task 2 – Creating the TestAccount Class

In this task, you complete the following steps to create a new `TestAccount` Java class.



Tool Reference – Java Development: Java Classes: Creating Java Classes

1. Create a Java class with the following characteristics:

Class Name: **TestAccount**

Project: **BankPrj**

Location: **Source Packages**

Package: **default package**

2. Add the main method to the `TestAccount` class:

```
public class TestAccount {
    public static void main(String[] args) {
        // code here
    }
}
```

Exercise 2: Creating a Test Program (Level 3)

3. Declare a variable of type `Account` and initialize that variable by creating an instance of the `Account` class with an initial balance of 100.

```
Account acct = new Account(100.0);
```

4. Use the `deposit` method to add 50 to the account.

```
acct.deposit(50.0);
```

5. Use the `withdraw` method to subtract 147 from the account.

```
acct.withdraw(147.0);
```

6. Use the `getBalance` method to retrieve the new account balance and use the `System.out.println` method to display the balance to the standard output stream.

```
System.out.println("Final account balance is " + acct.getBalance());
```

Task 3 – Compiling the `TestAccount` Class

Compile the `TestAccount` class, and make necessary changes to correct compilation errors.

Task 4 – Running the `TestAccount` Program

Run the `TestAccount` program. If there are runtime errors, make necessary changes to the `TestAccount` class, recompile it and run the program again.

The output of the `TestAccount` program should be similar to the following:

```
Final account balance is: 3.0
```

Exercise Summary



Discussion – Take a few minutes to discuss what experiences, issues, or discoveries you had during the lab exercise.

- Experiences

- Interpretations

- Conclusions

- Applications
