

# Coding Assignment 6

Source History

```
24 import javax.swing.SwingConstants;
```

Output

Debugger Console x CodingAssignment6 (run) x

```
ant -f C:\\Users\\frenc\\Documents\\NetBeansProjects\\CodingAssignment6 -Dnb.internal.action.name=run run
init:
Deleting: C:\\Users\\frenc\\Documents\\NetBeansProjects\\CodingAssignment6\\build\\built-jar.properties
deps-jar:
Updating property file: C:\\Users\\frenc\\Documents\\NetBeansProjects\\CodingAssignment6\\build\\built-jar.properties
compile:
run:
```

Building CodingAssignment6 (run)...

CodingAssignment6 (run) 40:1 INS | Windows (CR...

# Coding Assignment 6

Pick at least 4 related images. My app uses 4 characters from the cartoon series Flintstones.



Fred.png



Barney.png



Wilma.png



Betty.png

You can use .png or .jpeg or .gif files. Your file names should match the string you want to be entered as the guess for the character's name. Store these files in the same directory as your class files.

# Coding Assignment 6

**Create your Code6\_XXXXXXXXXX project.**

```
GUI - Label

package labeltest;

import javax.swing.JFrame;

public class LabelTest
{
    public static void main(String[] args)
    {
        LabelFrame labelFrame = new LabelFrame();
        labelFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        labelFrame.setSize(460, 380);
        labelFrame.setVisible(true);
    }
}
```

In `main()`, instantiate an object of the class `Password`.

Setup your `password` object the same way we set `labelFrame` in class `LabelTest` in the slides.

You will need to adjust the size as needed for your project.

This will be the only code that goes in `main()` and in class `Code6_XXXXXXXXXX`

```
public class TextFieldFrame extends JFrame
{
    private final JTextField textField1;
    private final JTextField textField2;
    private final JTextField textField3;
    private final JPasswordField passwordField;
```

# Coding Assignment 6

## Create a Password class file.

Class Password **extends** JFrame

Set up a `private final JPasswordField` variable like we did in class.

Create a variable to hold the user entered password.

Create a `final String` variable and set it to the actual password.

# Coding Assignment 6

## Create a Password class file.

Create the Password class constructor

- Call the superclass constructor
- set the layout to `FlowLayout`
- instantiate a `JPasswordField`
- look in the Java documentation on how to **set** the **E**cho **char** to 'X'
- add



```
TextFieldHandler handler = new TextFieldHandler();  
textField1.addActionListener(handler);  
textField2.addActionListener(handler);  
textField3.addActionListener(handler);  
passwordField.addActionListener(handler);
```

# Coding Assignment 6

## Add event handling to Password class

Instantiate an event handler from class `EventHandler`

Add an `ActionListener` to `passwordField`

# Coding Assignment 6

```
private class TextFieldHandler implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent event)
    {
        String string = "";

        if (event.getSource() == textField1)
            string = String.format("textField1: %s", event.getActionCommand());
        else if (event.getSource() == textField2)
            string = String.format("textField2: %s", event.getActionCommand());
        else if (event.getSource() == textField3)
            string = String.format("textField3: %s", event.getActionCommand());
        else if (event.getSource() == passwordField)
            string = String.format("passwordField: %s", event.getActionCommand());

        JOptionPane.showMessageDialog(null, string);
    }
}
```

## Add inner class EventHandler

private inner class EventHandler will implement interface ActionListener

If `event.getSource()` is `passwordField`, then set user entered password to `event.getActionCommand()`.

If user entered password equals the actual password, then pass `false` to `setVisible()` to hide the password entry box and set up `GameFrame` (next slide).

If user entered password does not equal the actual password, then use a `MessageDialog` box to display invalid password message.



# Coding Assignment 6

GUI - Label

```
package labeltest;

import javax.swing.JFrame;

public class LabelTest
{
    public static void main(String[] args)
    {
        JFrame labelFrame = new JFrame();
        labelFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        labelFrame.setSize(460, 380);
        labelFrame.setVisible(true);
    }
}
```

## Instantiating JFrame inside EventHandler

Set up JFrame like you did Password.

instantiate it

set default close operation

set size

set visible

Now create a new class file named JFrame.

# Coding Assignment 6

Create class `GameFrame`

`GameFrame` **extends** `JFrame`

Declare 4 `private final` variables

`JLabel label1`

`JButton OKButton`

`JButton CancelButton`

`JTextField textField1`

Declare a `String` to hold the character's name/answer to the question. I called mine `CCName`.

# Coding Assignment 6

## **GameFrame constructor**

Call super constructor, set the layout and set the default close operation.

Use a random number to choose one of your four character names. I used a `switch` with the random number to pick 1 of my 4. For example, a random number of 2 sets my `CCName` variable to "Wilma".

Use the randomly chosen name to create the `Icon` object.

```
Icon CC = new ImageIcon(getClass().getResource(CCName + ".png"));
```

# Coding Assignment 6

## GameFrame constructor continued...

```
public LabelFrame()
{
    super("Testing JLabel");
    setLayout(new FlowLayout());

    label1 = new JLabel("Label with text");
    label1.setToolTipText("This is label1");
    add(label1);

    Icon BB = new ImageIcon(getClass().getResource("BinaryBlaze.png"));
    label2 = new JLabel("Label with text and icon", BB, SwingConstants.LEFT);
    label2.setToolTipText("This is label2");
    add(label2);

    label3 = new JLabel();
    label3.setText("Label with icon and text at bottom");
    label3.setIcon(BB);
    label3.setHorizontalTextPosition(SwingConstants.CENTER);
    label3.setVerticalTextPosition(SwingConstants.BOTTOM);
    label3.setToolTipText("This is label3");
    add(label3);
}
```

Create a new `JLabel` called `label1`

- set the text
- set the icon
- set to horizontal and vertical position
- set the tool tip to be a hint
- add it

# Coding Assignment 6

## **GameFrame constructor continued...**

Instantiate an event handler from class `EventHandler` and name it `handler`

We'll create the inner class `EventHandler` shortly...

NOTE : all components in the `GameFrame` class will use the same event handler – do not create separate ones for the text field and the buttons.

# Coding Assignment 6

## GameFrame constructor continued...

```
public TextFieldFrame()
{
    super("Testing JTextField and JPasswordField");
    setLayout(new FlowLayout());

    textField1 = new JTextField(10);
    add(textField1);

    textField2 = new JTextField("Enter text here ");
    add(textField2);

    textField3 = new JTextField("Uneditable text field", 21);
    textField3.setEditable(false);
    add(textField3);

    passwordField = new JPasswordField("Hidden text");
    add(passwordField);

    TextFieldHandler handler = new TextFieldHandler();
    textField1.addActionListener(handler);
    textField2.addActionListener(handler);
    textField3.addActionListener(handler);
    passwordField.addActionListener(handler);
}
```

Instantiate a new `JTextField` named `textField1`.

Check the Java documentation on how to **select All** of the text in the field so that you can just start typing and overwrite it without having to manually select it or delete it.

Set an `ActionListener` for `textField1` and add `textField1` to the container.

# Coding Assignment 6

## GameFrame constructor continued and finished

```
public class ButtonFrame extends JFrame
{
    private final JButton plainButton;
    private final JButton fancyButton;

    public ButtonFrame()
    {
        super("Testing Buttons");
        setLayout(new FlowLayout());

        plainButton = new JButton("Plain Button");
        add(plainButton);

        Icon btn1 = new ImageIcon(getClass().getResource("BinaryBlaze.png"));
        Icon btn2 = new ImageIcon(getClass().getResource("Fred.png"));
        fancyButton = new JButton("Fancy Button", btn1);
        fancyButton.setRolloverIcon(btn2);
        add(fancyButton);

        ButtonHandler handler = new ButtonHandler();
        fancyButton.addActionListener(handler);
        plainButton.addActionListener(handler);
    }

    private class ButtonHandler implements ActionListener
    {
        @Override
        public void actionPerformed(ActionEvent event)
        {
            JOptionPane.showMessageDialog(ButtonFrame.this,
                String.format("You pressed %s", event.getActionCommand()));
        }
    }
}
```

Instantiate a new JButton named OKButton.

Set an ActionListener for OKButton and add OKButton to the container.

Instantiate a new JButton named CancelButton.

Set an ActionListener for CancelButton and add CancelButton to the container.

# Coding Assignment 6

```
private class TextFieldHandler implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent event)
    {
        String string = "";

        if (event.getSource() == textField1)
            string = String.format("textField1: %s", event.getActionCommand());
        else if (event.getSource() == textField2)
            string = String.format("textField2: %s", event.getActionCommand());
        else if (event.getSource() == textField3)
            string = String.format("textField3: %s", event.getActionCommand());
        else if (event.getSource() == passwordField)
            string = String.format("passwordField: %s", event.getActionCommand());

        JOptionPane.showMessageDialog(null, string);
    }
}
```

## Create inner class `EventHandler`

private inner class `EventHandler` will implement interface `ActionListener`

Set up a `String` that will hold the response that you will show to the user

Create a `boolean` that will be true if the user entered guess matches the character's name – initialize it to false



# Coding Assignment 6

```
private class TextFieldHandler implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent event)
    {
        String string = "";

        if (event.getSource() == textField1)
            string = String.format("textField1: %s", event.getActionCommand());
        else if (event.getSource() == textField2)
            string = String.format("textField2: %s", event.getActionCommand());
        else if (event.getSource() == textField3)
            string = String.format("textField3: %s", event.getActionCommand());
        else if (event.getSource() == passwordField)
            string = String.format("passwordField: %s", event.getActionCommand());

        JOptionPane.showMessageDialog(null, string);
    }
}
```

## Create inner class **EventHandler**

if the character's name equals\* the user entered string

set the response string to the guessed correctly phrase

set the `boolean` to true

else

set the response string to the guessed incorrectly phrase

\*be sure to ignore case

# Coding Assignment 6

```
private class TextFieldHandler implements ActionListener
{
    @Override
    public void actionPerformed(ActionEvent event)
    {
        String string = "";

        if (event.getSource() == textField1)
            string = String.format("textField1: %s", event.getActionCommand());
        else if (event.getSource() == textField2)
            string = String.format("textField2: %s", event.getActionCommand());
        else if (event.getSource() == textField3)
            string = String.format("textField3: %s", event.getActionCommand());
        else if (event.getSource() == passwordField)
            string = String.format("passwordField: %s", event.getActionCommand());

        JOptionPane.showMessageDialog(null, string);
    }
}
```

## Create inner class **EventHandler**

if `event.getSource()` is the `OKButton` or `textField1`  
show a `MessageDialog` box with the response string  
if your boolean is true

use `System.exit(0)` to shut down the program

else if `event.getSource()` is the `CancelButton`

use `System.exit(0)` to shut down the program

# Coding Assignment 6

Almost everything you need for the assignment is in the Swing slides. There are a few minor things I want you to find on your own in the online documentation in order to become familiar with it.

<https://docs.oracle.com/javase/7/docs/api/allclasses-noframe.html>

Google and Stack Overflow are NOT your friends for this assignment – they will lead you down a complicated path that will waste your time.

Stick with the simple things we did in class for this assignment.

# Coding Assignment 6

Create a standalone executable of your program by creating a `.jar` file.

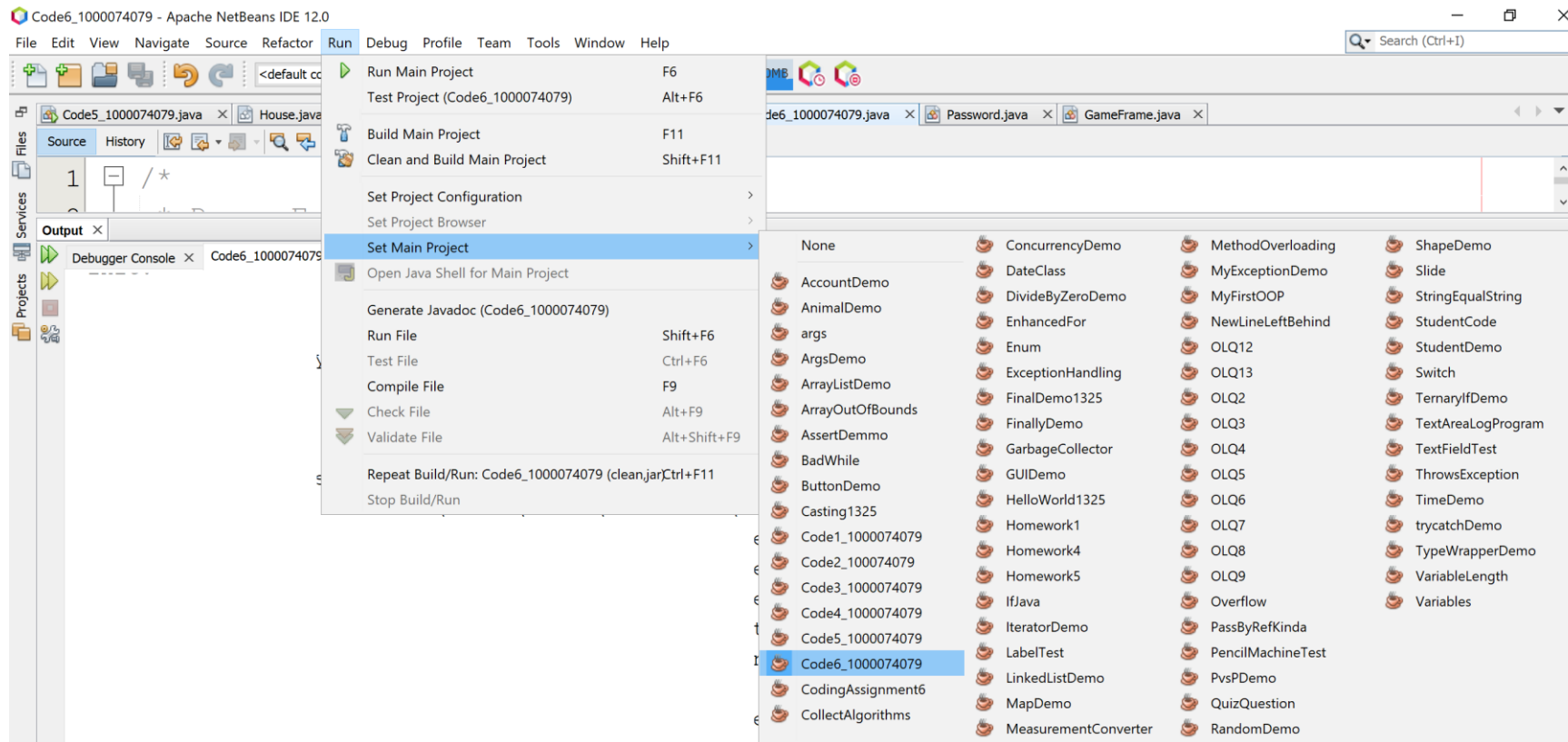
I have included the directions for creating a `.jar` file in NetBeans.

If you are using a different IDE, then look for similar terms.

Your code will be graded by running the executable file and looking at your code.

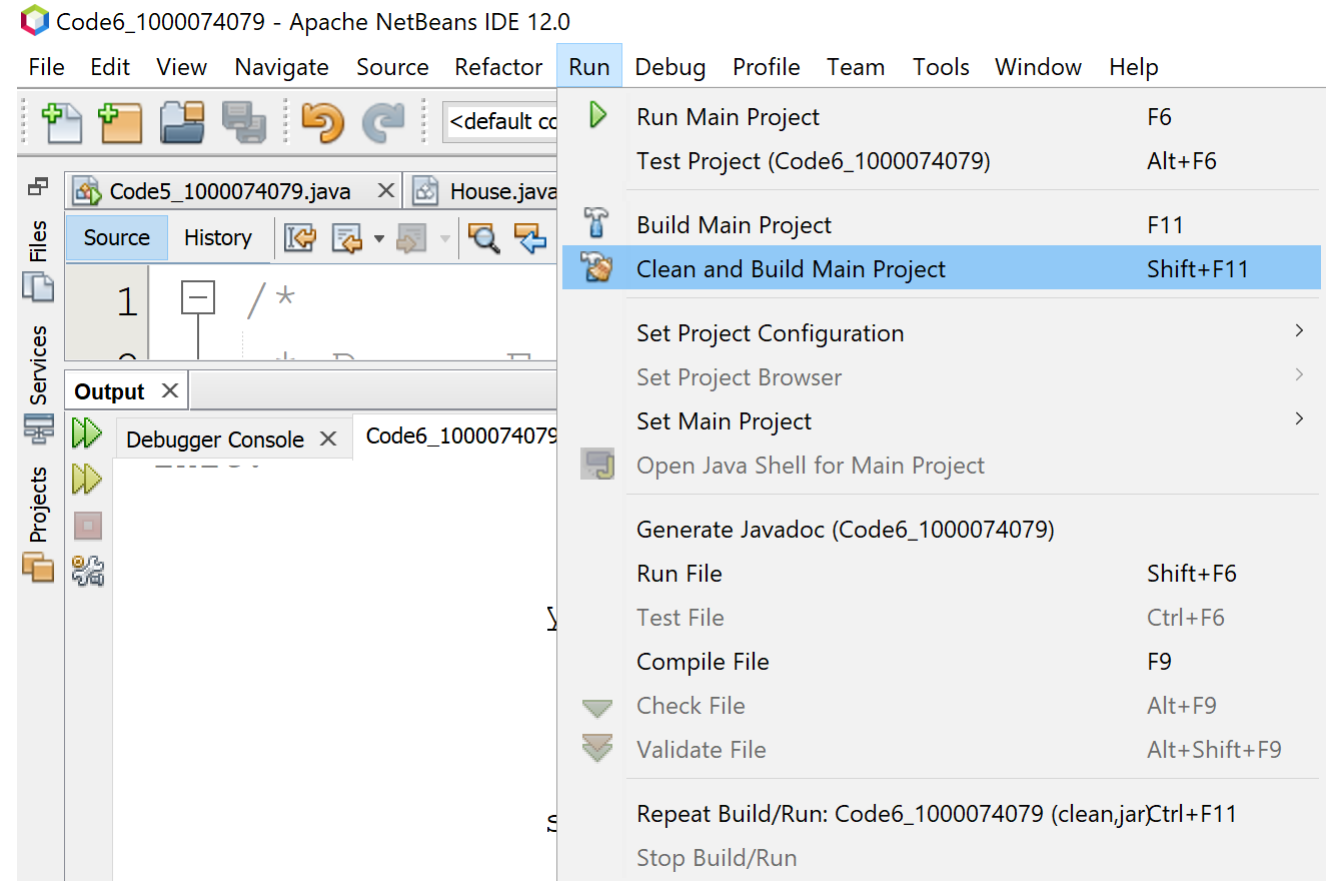
# Coding Assignment 6

Under Run, go to Set Main Project and make sure Code6\_xxxxxxxxxx is set as the Main Project.



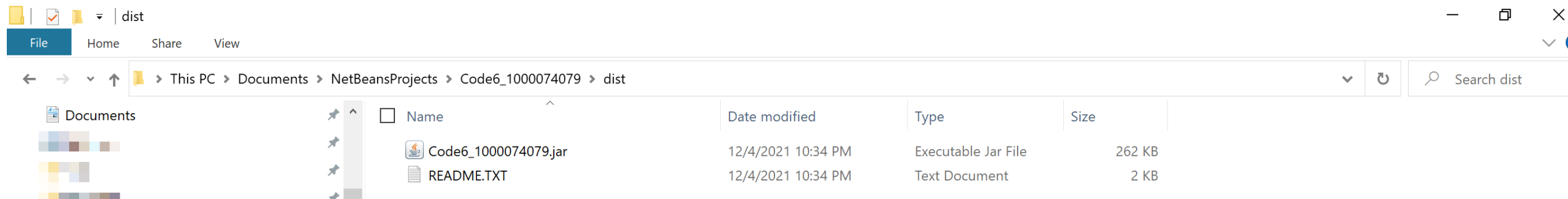
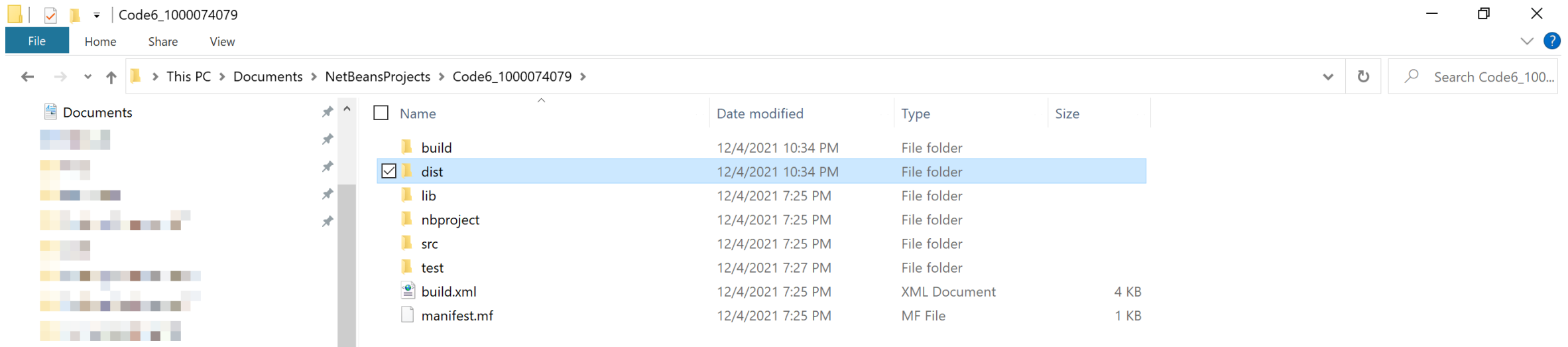
# Coding Assignment 6

Under Run, click on  
Clean and Build  
Main Project



# Coding Assignment 6

The `Code6_XXXXXXXXXX.jar` file will be created in a folder named `dist` in your NetBeans project folder.



You pick the password.

You pick the images.

You pick the question.

You pick the answer(s).

You pick the text of the `MessageDialog` boxes for a right and a wrong answer.

Your application needs to behave in the same way as the example in the assignment. You do not need to have the exact layout as my example. I used a vertical box and a horizontal box to control the placement of the components but you are not required to do this.

**Please be professional with your choices – no offensive images or offensive language.**