1) The programs `Account.java` and `AccountTest.java` are from your textbook. As stated in `AccountTest.java`, write the code to withdraw money from account.

**In Account.java**

```java
/* write code to declare method debit */
public void debit(double amount) {
    if(amount > getBalance()){
        System.out.println("Debit amount exceeded account balance");
    } else{
        balance = balance - amount;
    }
}
```

**In AccountTest.java**

```java
System.out.print( "Enter withdrawal amount for account1: " );
withdrawalAmount = input.nextDouble(); // obtain user input
System.out.printf("\nsubtracting %.2f from account1 balance\n", withdrawalAmount);
/* write code to withdraw money from account */
account1.debit(withdrawalAmount);

// display balances
System.out.printf( "account1 balance: $%.2f\n", account1.getBalance() );
System.out.printf( "account2 balance: $%.2f\n", account2.getBalance() );

System.out.print( "Enter withdrawal amount for account2: " );
withdrawalAmount = input.nextDouble(); // obtain user input
System.out.printf("\nsubtracting %.2f from account2 balance\n", withdrawalAmount);
/* write code to withdraw from account */
account2.debit(withdrawalAmount);
```
I have successfully completed the task. Score:10/10

2) Program templates Palindrome.java and PalindromeTest.java are provided. Write the code as stated in Palindrome.java and test the code using PalindromeTest.java

Palindrom.java

```java
import java.util.Scanner;

public class Palindrome {
    // checks if a 5-digit number is a palindrome
    public void checkPalindrome() {
        Scanner input = new Scanner(System.in);

        int number; // user input number
        int digit1; // first digit
        int digit2; // second digit
        int digit4; // fourth digit
        int digit5; // fifth digit
        int digits; // number of digits in input

        number = 0;
        digits = 0;

        /* Write code that inputs a five-digit number. Display an error message
         * if the number is not five digits. Loop until a valid input is received. */
        //prompts for User Input
        System.out.println("Enter a five digit number"); // prompt
        number = input.nextInt();
        while (number < 9999 || number > 99999) //if the number is not five digits
        {
            System.out.println("The number you entered is not a five-digit number. \n Please
            Enter a five digit number"); //Display an error message
            number = input.nextInt();
        }
        /* Write code that separates the digits in the five digit number. Use
         * division to isolate the left-most digit in the number, use a remainder
         * calculation to remove that digit from the number. Then repeat this
         * process. */
        digit1 = number/10000; //
        digit2 = (number%10000)/1000;
        digit4 = (number%100)/10;
        digit5 = number%10;

        /* Write code that determines whether the first and last digits are
         * identical and the second and Fourth digits are identical. Output
         * whether or not the original string is a palindrome. */
        if(digit1 == digit5 && digit2==digit4) //tests whether the first and last digits are
        identical and the second and Fourth digits are identical.
            System.out.println(number + " is a palindrome!"); //Output whether its a palindrome
        else
```
I have successfully completed the task. Score: 10/10

3) Program templates Largest.java and LargestTest.java are provided.
Write the code as stated in Largest.java and test the code using LargestTest.java

Largest.java

```java
import java.util.Scanner;

public class Largest
{
    // determine the largest of 10 numbers
    public void determineLargest()
    {
        Scanner input = new Scanner( System.in );

        int largest; // largest number
        int number; // user input
        int counter; // number of values entered

        /* write code to get the first integer and store it in variable largest */
        System.out.println("Enter a number:");
        largest = input.nextInt();

        /* write code to initialize the number of integers entered */
        counter = 1;

        /* write code to loop until 10 numbers are entered */
        /* write code to prompt the user to enter a number and read that number */
        while (counter < 10)
        {
            System.out.println("Enter a another number:");
            number = input.nextInt();
            System.out.printf("\nNumber entered : %d\n", number);

            /* write code to test whether the number entered is greater than the largest
            if so, replace the value of largest with the entered number */
    
```
if(number > largest)  
{  
    largest = number;  
}

/* write code to increment the number of integers entered */  
counter++;  
} //end while loop

System.out.printf("Largest number is %d\n", largest);  
} // end method determineLargest  
} // end class Largest

I have successfully completed the task. Score: 10/10

4) Modify DrawPanel.java of Fig. 4.18 to draw the following:

Triangle
import java.awt.Graphics;  
import javax.swing.JPanel;

public class DrawPanel extends JPanel  
{
    // draws a ellipse  
    public void paintComponent( Graphics g )
    {
        // call paintComponent to ensure the panel displays correctly  
        super.paintComponent( g );

        // draws ellipse  
        g.drawOval( 150, 25, 150, 75);
Fig. 4.18: DrawPanel.java

// Using drawLine to make a triangle.
import java.awt.Graphics;
import javax.swing.JPanel;

public class DrawPanel extends JPanel
{
    // draws an Triangle from the corners of the panel
    public void paintComponent( Graphics g )
    {
        // call paintComponent to ensure the panel displays correctly
        super.paintComponent( g );

        // draw left leg of triangle
        g.drawLine( 50, 100, 100, 0 );

        // draw a right leg of triangle
        g.drawLine( 100, 0, 150, 100 );

        // draw bottom line of triangle
        g.drawLine( 50, 100, 150, 100 );
    }
} // end class DrawPanel

CIRCLE

import java.awt.Graphics;
import javax.swing.JPanel;

public class DrawPanel extends JPanel
{
    // draws an Triangle from the corners of the panel
    public void paintComponent( Graphics g )
    {
        // call paintComponent to ensure the panel displays correctly
        super.paintComponent( g );

        int width = getWidth(); // total width
        int height = getHeight(); // total height

        // draw left line
        g.drawLine( 50, 0, 100, 0 );

        // draw a right line
        g.drawLine( 100, 0, 150, 0 );

        // draw bottom line
        g.drawLine( 50, 0, 100, 0 );
    }
} // end class DrawPanel
import java.awt.Graphics;
import javax.swing.JPanel;

public class DrawPanel extends JPanel {

    // draws a Pentagon
    public void paintComponent( Graphics g )
    {
        // call paintComponent to ensure the panel displays correctly
        super.paintComponent( g );

        // draws top of pentagon
        g.drawLine( 50, 50 , 100, 0 );

        g.drawLine( 100 , 0, 150, 50 );

        // draw left leg of pentagon
        g.drawLine( 50, 50 , 50 , 125);
    }
}

Pentagon

public class DrawPanel extends JPanel
{

    // draws a pentagon
    public void paintComponent( Graphics g )
    {
        // call paintComponent to ensure the panel displays correctly
        super.paintComponent( g );

        int width = getWidth(); // total width
        int height = getHeight(); // total height

        // draws top of pentagon
        g.drawLine( 50, 50, 100, 0 );
        g.drawLine( 100, 0, 150, 50 );

        // draw left leg of pentagon
        g.drawLine( 50, 50, 50 , 125);

        // draw right leg of pentagon
        g.drawLine( 150, 50, 150, 125);
    }
}

Triangle

import java.awt.Graphics;
import javax.swing.JPanel;

public class DrawPanel extends JPanel
{
    // draws a filled in rectangle
    public void paintComponent( Graphics g )
    {
        // call paintComponent to ensure the panel displays correctly
        super.paintComponent( g );

        // fills in rectangle
        int y=50;
        while(y <= 125)
        {
            g.drawLine( 50, y, 150, y);
            y += 1;
        }

        //draw left leg of rectangle
        g.drawLine( 50, 50, 50 , 125);

        //draw right leg of rectangle
        g.drawLine( 150, 50, 150, 125);
import java.awt.Graphics;
import javax.swing.JPanel;

public class DrawPanel extends JPanel {

    // draws a ellipse
    public void paintComponent( Graphics g )
    {
        // call paintComponent to ensure the panel displays correctly
        super.paintComponent( g );

        // draws ellipse
        g.drawOval( 150, 25, 150, 75);
    }

} // end class DrawPanel