Account.java

```java
public class Account {
    private double balance; // instance variable that stores the balance

    // constructor
    public Account( double initialBalance ) {
        // validate that initialBalance is greater than 0.0;
        // if it is not, balance is initialized to the default value 0.0
        if ( initialBalance > 0.0 )
            balance = initialBalance;
    } // end Account constructor

    // credit (add) an amount to the account
    public void credit( double amount ) {
        balance = balance + amount; // add amount to balance
    } // end method credit

    // withdraw (subtract) an amount to the account
    public void withdraw( double amount) {
        balance = balance - amount; // subtract amount to balance
    } // end method withdraw

    // return the account balance
    public double getBalance() {
        return balance; // gives the value of balance to the calling method
    } // end method getBalance
}
// end class Account
```

AccountTest.java

```java
import java.util.Scanner;

public class AccountTest {
    // main method begins execution of Java application
    public static void main( String[] args ) {
        Account account1 = new Account( 50.00 ); // create Account object
        Account account2 = new Account( -7.53 ); // create Account object

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

        // create Scanner to obtain input from command window
        Scanner input = new Scanner( System.in );
    }
}
```
double depositAmount; // deposit amount read from user
double withdrawAmount; // withdraw amount read from user

System.out.println( "Enter deposit amount for account1: " ); // prompt
depositAmount = input.nextDouble(); // obtain user input
System.out.printf( "\nadding %.2f to account1 balance\n", depositAmount );
account1.credit( depositAmount ); // add to account1 balance

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

System.out.println( "Enter deposit amount for account2: " ); // prompt
depositAmount = input.nextDouble(); // obtain user input
System.out.printf( "\nadding %.2f to account2 balance\n", depositAmount );
account2.credit( depositAmount ); // add to account2 balance

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

System.out.println( "Enter withdraw amount for account1: " ); // prompt
withdrawAmount = input.nextDouble(); // obtain user input
System.out.printf( "\nsubtracting %.2f to account1 balance\n", withdrawAmount );
account1.withdraw( withdrawAmount ); // subtract from account1 balance

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

System.out.println( "Enter withdraw amount for account2: " ); // prompt
withdrawAmount = input.nextDouble(); // obtain user input
System.out.printf( "\nsubtracting %.2f to account2 balance\n", withdrawAmount );
account2.withdraw( withdrawAmount ); // subtract from account2 balance

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

} // end main
} // end class AccountTest
I finished the task successfully.

```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 digit3; // third 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. */
        System.out.print("Please enter 5-digit number: ");
        number = input.nextInt();

        while ( 100000 < number || number < 9999)
        {
```

Palindrome.java

```java
```
You have not entered a 5-digit number please try again

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 % 10;
number = number / 10;
digit2 = number % 10;
number = number / 10;
digit3 = number % 10;
number = number / 10;
digit4 = number % 10;
number = number / 10;
digit5 = number % 10;
System.out.printf("Digit 1: %d", digit1);
System.out.printf("Digit 2: %d", digit2);
System.out.printf("Digit 3: %d", digit3);
System.out.printf("Digit 4: %d", digit4);
System.out.printf("Digit 5: %d", digit5);
/* 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)
    System.out.println("Digit is a palindrome");
else
    System.out.println("Digit is not a palindrome");
} // end method checkPalindrome
} // end class Palindrome

I finished the task successfully.
Largest.java

import java.util.Scanner;

public class Largest
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.print("Enter largest number: ");
    largest = input.nextInt();

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

    /* write code to loop until 10 numbers are entered */
    while (counter < 10)
    {
        /* write code to prompt the user to enter a number and read that number */
        System.out.print("\nEnter another number: ");
        number = input.nextInt();
        System.out.printf("You have just entered %d", 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++;
    }
    System.out.printf("\nLargest number is %d\n", largest);
} // end method determineLargest
} // end class Largest

I finish the task successfully.
DrawPanel.java

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

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

        int x1 = 200; // total width
        int y1 = 200; // total height
        int x2 = 0;
        int y2 = 200;
        int x3 = x1/2;
        int y3 = 0;

        // draw a line from the upper-left to the lower-right
g.drawLine(x1, y1, x2, y2); // base of triangle
g.drawLine(x1, y1, x3, y3); // right side
g.drawLine(x3, y3, x2, y2); // left side

} // end method paintComponent
} // end class DrawPanel

b) Circle

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

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

        int x = 50;
        int y = 50;
        int width = 100;
        int height = 100;

        // draw a line from the upper-left to the lower-right
g.drawOval(x, y, width, height);

}  // end method paintComponent
}  // end class DrawPanel

c)  Pentagon

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

public class DrawPanel extends JPanel{

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

        int x1 = 200; // total width
        int y1 = 200; // total height
        int x2 = 0;
        int y2 = 200;
        int x3 = x1/2;
        int y3 = 0;
        int x4 = 0;
        int y4 = 80;
        int x5 = 200;
        int y5 = 80;

        // draw a line from the upper-left to the lower-right
        g.drawLine(x1, y1, x2, y2); // base of triangle
        g.drawLine(x1, y1, x5, y5); // right side
d) Filled rectangle

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

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

        int x4 = 0;
        int y = 80;
        int x5 = 200;

        // draw a line from the upper-left to the lower-right
        while ( y <= 200 )
        {
            g.drawLine(x4, y, x5, y); // top side
            y = y + 1;
        }
    }
}
```
import java.awt.Graphics;
import javax.swing.JPanel;

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

        int x = 50;
        int y = 50;
        int width = 175;
        int height = 100;

        // draw a line from the upper-left to the lower-right
        g.drawOval(x, y, width, height);
    }
}

// end class DrawPanel