الاسم : زهرة خالد

تقنية معلومات عربي

Review of Java programming basics

programs

- Hello world program

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

- Print numbers from 1-10

```
public class PrintNumbers {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
    }
}
```

- Print array elements

```
public class PrintArray {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5};

        for (int i = 0; i < numbers.length; i++) {
            System.out.println(numbers[i]);
        }
    }
}
```

الاسم : زهرة خالد

تقنية معلومات عربي

- Input array elements

```
import java.util.Scanner;


public class InputArray {
    public static void main(String[] args) {
        int[] numbers = new int[5];
        Scanner scanner = new Scanner(System.in);


        System.out.println("Enter 5 numbers:");
        for (int i = 0; i < numbers.length; i++) {
            numbers[i] = scanner.nextInt();
        }


        for (int i = 0; i < numbers.length; i++) {
            System.out.println(numbers[i]);
        }
    }
}
```

- Define method to print array elements

```
public class ArrayUtils {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5};
        printArray(numbers);
    }


    public static void printArray(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.println(arr[i]);
        }
    }
}
```

- Define method to Input array elements

```
import java.util.Scanner;

public class ArrayUtils {
    public static void main(String[] args) {
        int[] numbers = new int[5];
        inputArray(numbers);
        printArray(numbers);
    }

    public static void inputArray(int[] arr) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter 5 numbers:");
        for (int i = 0; i < arr.length; i++) {
            arr[i] = scanner.nextInt();
        }
    }

    public static void printArray(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.println(arr[i]);
        }
    }
}
```

- Array of objects (Students)

```
public class Student {
```

الاسم : زهرة خالد

تقنية معلومات عربي

```
private String name;

public Student(String name) {
    this.name = name;
}

public String getName() {
    return name;
}

public static void main(String[] args) {
    Student[] students = new Student[3];
    students[0] = new Student("Alice");
    students[1] = new Student("Bob");
    students[2] = new Student("Charlie");

    for (int i = 0; i < students.length; i++) {
        System.out.println(students[i].getName());
    }   }
```

الاسم : زهرة خالد

تقنية معلومات عربي

R-1.1	Write a short Java method, inputAllBaseTypes, that inputs a different value of each base type from the standard input device and prints it back to the standard output device.
	```
	import java.util.Scanner;

	public class InputBaseTypes {
	    public static void main(String[] args) {
	        inputAllBaseTypes();
	    }

	    public static void inputAllBaseTypes() {
	        Scanner scanner = new Scanner(System.in);

	        System.out.print("Enter a boolean value: ");
	        boolean boolValue = scanner.nextBoolean();
	        System.out.println("Boolean value: " + boolValue);

	        System.out.print("Enter a byte value: ");
	        byte byteValue = scanner.nextByte();
	        System.out.println("Byte value: " + byteValue);

	        System.out.print("Enter a short value: ");
	        short shortValue = scanner.nextShort();
	        System.out.println("Short value: " + shortValue);

	        System.out.print("Enter an int value: ");
	        int intValue = scanner.nextInt();
	        System.out.println("Int value: " + intValue);

	        System.out.print("Enter a long value: ");
	        long longValue = scanner.nextLong();
	        System.out.println("Long value: " + longValue);

	        System.out.print("Enter a float value: ");
	        float floatValue = scanner.nextFloat();
	        System.out.println("Float value: " + floatValue);

	        System.out.print("Enter a double value: ");
	        double doubleValue = scanner.nextDouble();
	        System.out.println("Double value: " + doubleValue);

	        System.out.print("Enter a character: ");
	        char charValue = scanner.next().charAt(0);
	        System.out.println("Character value: " + charValue);

	        scanner.close();
	```

	} }

R-1.2	Suppose that we create an array A of GameEntry objects, which has an integer scores field, and we clone A and store the result in an array B. If we then immediately set A[4].score equal to 550, what is the score value of the GameEntry object referenced by B[4]?
	
	B[4] =score
	A[4] =score

R-1.3	Write a short Java method, isMultiple, that takes two long values, n and m, and returns true if and only if n is a multiple of m, that is, n = mi for some integer i.
	
	public class Main {
	    public static boolean isMultiple(long n, long m) {
	        return n % m == 0;
	    }
	
	    public static void main(String[] args) {
	        long n = 12;
	        long m = 3;
	        System.out.println(isMultiple(n, m)); // Output: true
	
	        n = 17;
	        m = 5;
	        System.out.println(isMultiple(n, m)); // Output: false
	    }
	}

R-1.4	Write a short Java method, isEven, that takes an int i and returns true if and only if i is even. Your method cannot use the multiplication, modulus, or division operators, however.
	public class Main {
	    public static void main(String[] args) {
	        System.out.println(isEven(4)); // prints: true
	        System.out.println(isEven(5)); // prints: false
	    }
	
	    public static boolean isEven(int i) {
	        // i = 0 is considered as even
	        if (i == 0) {
	            return true;
	        }
	
	        // Subtract 1 from i and recursively call the isEven method

```
        return isEven(i - 1);
    }
}
```

R-1.5 Write a short Java method that takes an integer n and returns the sum of all positive integers less than or equal to n.

```
public class SumOfIntegers {
    public static void main(String[] args) {
        int n = 10;
        System.out.println("The sum of integers from 1 to " + n + " is: " +
sumWithoutOperators(n));
    }

    public static int sumWithoutOperators(int n) {
        if (n == 1) {
            return 1;
        }
        return n + sumWithoutOperators(n - 1);
    }
}
```

R-1.6 Write a short Java method that takes an integer n and returns the sum of all the odd positive integers less than or equal to n.

```
public class SumOfOddIntegers {
    public static void main(String[] args) {
        int n = 10;
        System.out.println("The sum of odd integers from 1 to " + n + " is: " +
sumWithoutOperators(n));
    }

    public static int sumWithoutOperators(int n) {
        if (n == 1) {
            return 1;
        }
        return n + sumWithoutOperators(n - 2);
    }
}
```

R-1.7 Write a short Java method that takes an integer n and returns the sum of the squares of all positive integers less than or equal to n.

```
public class Main {
    public static void main(String[] args) {
        System.out.println(sumSquares(5)); // prints 55
    }
    public static int sumSquares(int n) {
        int sum = 0;
        for (int i = 1; i <= n; i++) {
            sum += i * i;
        }
        return sum;
    }
}
```

R-1.8

Write a short Java method that counts the number of vowels in a given character string.

```
public class Main {
    public static void main(String[] args) {
        System.out.println(countVowels("Hello World")); // prints 3
    }

    public static int countVowels(String str) {
        int count = 0;
        String lowerCase = str.toLowerCase();
        for (int i = 0; i < lowerCase.length(); i++) {
            char c = lowerCase.charAt(i);
            if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') {
                count++;
            }
        }
        return count;
    }
}
```

R-1.9

Write a short Java method that uses a StringBuilder instance to remove all the punctuation from a string s storing a sentence, for example, transforming the string "Let’s try, Mike!" to "Lets try Mike".

```
public class Main {
    public static void main(String[] args) {
        System.out.println(removePunctuation("Let’s try, Mike!")); // prints "Lets try
Mike"
    }

    public static String removePunctuation(String s) {
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < s.length(); i++) {
            char c = s.charAt(i);
            if (!Character.isPunctuation(c)) {
                sb.append(c);
            }
        }
        return sb.toString();
    }
}
```

R-1.10

Write a Java class, Flower, that has three instance variables of type String, int, and float, which respectively represent the name of the flower, its number of petals, and price. Your class must include a constructor method that initializes each variable to an appropriate value, and your class should include methods for setting the value of each type, and getting the value of each type.

```
public class Flower {
    private String name;
    private int numberOfPetals;
    private float price;

    public Flower(String name, int numberOfPetals, float price) {
        this.name = name;
        this.numberOfPetals = numberOfPetals;
        this.price = price;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getNumberOfPetals() {
        return numberOfPetals;
    }

    public void setNumberOfPetals(int numberOfPetals) {
        this.numberOfPetals = numberOfPetals;
    }

    public float getPrice() {
        return price;
    }

    public void setPrice(float price) {
        this.price = price;
    }
}
```

الاسم : زهرة خالد

تقنية معلومات عربي

```
public class Main {
   public static void main(String[] args) {
      Flower flower = new Flower("Rose", 50, 50.5f);
      System.out.println(flower.getName());
      System.out.println(flower.getNumberOfPetals());
      System.out.println(flower.getPrice());

      flower.setName("Tulip");
      flower.setNumberOfPetals(30);
      flower.setPrice(40.4f);
      System.out.println(flower.getName());
System.out.println(flower.getNumberOfPetals());
      System.out.println(flower.getPrice());
   }
}
```

R-1.11

Modify the CreditCard class from Code Fragment 1.5 to include a method that updates the credit limit.

```
public class CreditCard {
   private String accountNumber;
   private String customerName;
   private double creditLimit;

   public CreditCard(String accountNumber, String customerName, double
creditLimit) {
      this.accountNumber = accountNumber;
      this.customerName = customerName;
      this.creditLimit = creditLimit;
   }

   public String getAccountNumber() {
      return accountNumber;
   }

   public String getCustomerName() {
      return customerName;
   }

   public double getCreditLimit() {
      return creditLimit;
   }

   public void updateCreditLimit(double newCreditLimit) {
      creditLimit = newCreditLimit;
   }
}
```

الاسم : زهرة خالد

تقنية معلومات عربي

R-1.12	Modify the CreditCard class from Code Fragment 1.5 so that it ignores any request to process a negative payment amount.
	`public class CreditCard {`
	    `private String accountNumber;`
	    `private String customerName;`
	    `private double creditLimit;`
	
	    `public CreditCard(String accountNumber, String customerName, double creditLimit) {`
	        `this.accountNumber = accountNumber;`
	        `this.customerName = customerName;`
	        `this.creditLimit = creditLimit;`
	    `}`
	
	    `public String getAccountNumber() {`
	        `return accountNumber;`
	    `}`
	
	    `public String getCustomerName() {`
	        `return customerName;`
	    `}`
	
	    `public double getCreditLimit() {`
	        `return creditLimit;`
	    `}`
	
	    `public void updateCreditLimit(double newCreditLimit) {`
	        `creditLimit = newCreditLimit;`
	    `}`
	
	    `public void processPayment(double paymentAmount) {`
	        `if (paymentAmount >= 0) {`
	            `creditLimit -= paymentAmount;`
	        `}`
	    `}`
	`}`

R-1.13	Modify the declaration of the first for loop in the main method in Code Fragment 1.6 so that its charges will cause exactly one of the three credit cards to attempt to go over its credit limit. Which credit card is it?
	
	`public class Main {`
	    `public static void main(String[] args) {`
	        `CreditCard[] wallet = new CreditCard[3];`
	        `wallet[0] = new CreditCard("John Bowman", "California Savings", "5391 0375 9387 5309", 5000);`

الاسم : زهرة خالد

تقنية معلومات عربي

```
        wallet[1] = new CreditCard("John Bowman", "California Federal", "3485 0399
3395 1954", 3500);
        wallet[2] = new CreditCard("John Bowman", "California Finance", "6011 4902
3294 2994", 2500);

        for (int val = 1; val <= 16; val++) {
            wallet[2].charge(225); // Increase charging amount for the third credit card
            wallet[1].charge(175);
            wallet[0].charge(125);
        }

        for (CreditCard card : wallet) {
            System.out.println(card);
            while (card.getBalance() > 200.0) {
                card.makePayment(200);
                System.out.println("New balance = " + card.getBalance());
            }
        }
    }
}
```