

## Review of basic Java programming

programs

- Hello world program
- Print numbers from 1-10
- Print array elements
- Input array elements
- Define method to print array elements
- Define method to Input array elements
- Array of objects (Students)

## Homework

Create a github account and upload lecture programs to your account

1	R-1.1	<p>Write a short Java method, <code>inputAllBaseTypes</code>, that inputs a different value of each base type from the standard input device and prints it back to the standard output device.</p> <p><code>import java.util.Scanner;</code></p> <pre> public class InputAllBaseTypes {     public static void inputAllBaseTypes() {         Scanner scanner = new Scanner(System.in);          System.out.print("Enter a byte value: ");         byte byteValue = scanner.nextByte();          System.out.print("Enter a short value: ");         short shortValue = scanner.nextShort();          System.out.print("Enter an int value: ");         int intValue = scanner.nextInt();          System.out.print("Enter a long value: ");         long longValue = scanner.nextLong();          System.out.print("Enter a float value: "); </pre>
---	-------	--

		<pre> float floatValue = scanner.nextFloat();  System.out.print("Enter a double value: "); double doubleValue = scanner.nextDouble();  System.out.print("Enter a char value: "); char charValue = scanner.next().charAt(0);  System.out.print("Enter a boolean value: "); boolean booleanValue = scanner.nextBoolean();  System.out.println("You entered:"); System.out.println("Byte: " + byteValue); System.out.println("Short: " + shortValue); System.out.println("Int: " + intValue); System.out.println("Long: " + longValue); System.out.println("Float: " + floatValue); System.out.println("Double: " + doubleValue); System.out.println("Char: " + charValue); System.out.println("Boolean: " + booleanValue);  scanner.close(); }  public static void main(String[] args) {     inputAllBaseTypes(); } </pre>
2	R-1.2	<p>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]?</p> <p>لأنها تشير إلى نفس الكائن 550 ستصبح B[4].score القيمة</p>

3	R-1.3	<p>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, <math>n = m_i</math> for some integer i.</p> <pre> public class IsMultiple {     public static boolean isMultiple(long n, long m) {         return (n % m == 0);     }      public static void main(String[] args) {         System.out.println(isMultiple(10, 2)); // true         System.out.println(isMultiple(10, 3)); // false     } } </pre>
4	R-1.4	<p>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.</p> <pre> public class IsEven {     public static boolean isEven(int i) {         return (i &amp; 1) == 0; // باستخدام العمليات على الـ bits     }      public static void main(String[] args) {         System.out.println(isEven(4)); // true         System.out.println(isEven(7)); // false     } } </pre>
5	R-1.5	<p>Write a short Java method that takes an integer n and returns the sum of all positive integers less than or equal to n.</p> <pre> public class SumUpToN {     public static int sumUpToN(int n) {         int sum = 0;         for (int i = 1; i &lt;= n; i++) {             sum += i;         }         return sum;     }      public static void main(String[] args) { </pre>

		<pre>         System.out.println(sumUpToN(5)); // 15     } } </pre>
6	R-1.6	<p>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.</p> <pre> public class SumOddUpToN {     public static int sumOddUpToN(int n) {         int sum = 0;         for (int i = 1; i &lt;= n; i += 2) {             sum += i;         }         return sum;     }      public static void main(String[] args) {         System.out.println(sumOddUpToN(5)); // 9     } } </pre>
7	R-1.7	<p>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.</p> <pre> public class SumSquaresUpToN {     public static int sumSquaresUpToN(int n) {         int sum = 0;         for (int i = 1; i &lt;= n; i++) {             sum += i * i;         }         return sum;     }      public static void main(String[] args) {         System.out.println(sumSquaresUpToN(3)); // 14     } } </pre>
8	R-1.8	<p>Write a short Java method that counts the number of vowels in a given character string.</p> <pre> public class CountVowels {     public static int countVowels(String str) { </pre>

		<pre> int count = 0; str = str.toLowerCase(); for (char c : str.toCharArray()) {     if ("aeiou".indexOf(c) != -1) {         count++;     } } return count; }  public static void main(String[] args) {     System.out.println(countVowels("Hello World")); // 3 } </pre>
9	R-1.9	<p>Write a short Java method that uses a <code>StringBuilder</code> instance to remove all the punctuation from a string <code>s</code> storing a sentence, for example, transforming the string "Let's try, Mike!" to "Lets try Mike".</p> <pre> public class RemovePunctuation {     public static String removePunctuation(String str) {         return str.replaceAll("\\p{Punct}", "");     }      public static void main(String[] args) {         System.out.println(removePunctuation("Let's try, Mike!"));         // Lets try Mike     } } </pre>
10	R-1.10	<p>Write a Java class, <code>Flower</code>, that has three instance variables of type <code>String</code>, <code>int</code>, and <code>float</code>, 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.</p> <pre> public class Flower {     private String name;     private int petals;     private float price;      public Flower(String name, int petals, float price) { </pre>

		<pre>         this.name = name;         this.petals = petals;         this.price = price;     }      public String getName() {         return name;     }      public void setName(String name) {         this.name = name;     }      public int getPetals() {         return petals;     }      public void setPetals(int petals) {         this.petals = petals;     }      public float getPrice() {         return price;     }      public void setPrice(float price) {         this.price = price;     } } </pre>
11	R-1.11	<p>Modify the CreditCard class from Code Fragment 1.5 to include a method that updates the credit limit.</p> <pre> public class CreditCard {     private double creditLimit;      public void updateCreditLimit(double newLimit) {         this.creditLimit = newLimit;     } } </pre>

12	R-1.12	<p>Modify the CreditCard class from Code Fragment 1.5 so that it ignores any request to process a negative payment amount.</p> <pre> public class CreditCard {     public void processPayment(double payment) {         if (payment &lt; 0) {             System.out.println("Negative payment ignored.");             return;         }         // عملية الدفع هنا     } } </pre>
13	R-1.13	<p>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?</p> <pre> for (int i = 1; i &lt;= charges.length; i++) {     charges[i] += someValue;     if (charges[i] &gt; creditLimit) {         System.out.println("Card " + i + " exceeded limit.");         break;     } } </pre>