

First assignment

Data structures and Algorithms
Atheer Ghaberi



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> <pre>import java.util.Scanner; public class BaseTypeInput { public static void inputAllBaseTypes() { Scanner scanner = new Scanner(System.in); System.out.print("Enter an integer: "); int intValue = scanner.nextInt(); System.out.println("You entered integer: " + intValue); System.out.print("Enter a double: "); double doubleValue = scanner.nextDouble(); System.out.println("You entered double: " + doubleValue); System.out.print("Enter a boolean (true/false): ");</pre>
---	-------	---

		<pre> boolean booleanValue = scanner.nextBoolean(); System.out.println("You entered boolean: " + booleanValue); System.out.print("Enter a character: "); char charValue = scanner.next().charAt(0); // Read the next token and get the first character System.out.println("You entered character: " + charValue); System.out.print("Enter a string: "); String stringValue = scanner.next(); // Read the next token as a String System.out.println("You entered string: " + stringValue); scanner.close(); // Close the scanner } 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>the score value of the GameEntry object referenced by B[4] after the modification will be 550.</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, $n = mi$ for some integer i.</p> <pre> public class MultipleChecker { public static boolean isMultiple(long n, long m) { if (m == 0) { return false; // n is not a multiple of zero } return n % m == 0; // Check if the remainder is zero } public static void main(String[] args) { long n = 20; long m = 5; System.out.println("Is " + n + " a multiple of " + m + "? " + isMultiple(n, m)); // true n = 21; m = 5; System.out.println("Is " + n + " a multiple of " + m + "? " + isMultiple(n, m)); // 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 EvenChecker { public static boolean isEven(int i) { return (i & 1) == 0; } } </pre>

		<pre> public static void main(String[] args) { System.out.println("Is 4 even? " + isEven(4)); // true System.out.println("Is 7 even? " + isEven(7)); // false System.out.println("Is 0 even? " + isEven(0)); // true System.out.println("Is -2 even? " + isEven(-2)); // true System.out.println("Is -3 even? " + isEven(-3)); // 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 SumCalculator { public static int sumOfPositiveIntegers(int n) { if (n < 1) { return 0; // Return 0 if n is less than 1 } int sum = 0; for (int i = 1; i <= n; i++) { sum += i; // Add each integer to the sum } return sum; } public static void main(String[] args) { System.out.println("Sum of positive integers up to 5: " + sumOfPositiveIntegers(5)); // 15 System.out.println("Sum of positive integers up to 10: " + sumOfPositiveIntegers(10)); // 55 System.out.println("Sum of positive integers up to 0: " + sumOfPositiveIntegers(0)); // 0 System.out.println("Sum of positive integers up to -3: " + sumOfPositiveIntegers(-3)); // 0 } } </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 OddSumCalculator { public static int sumOfOddPositiveIntegers(int n) { if (n < 1) { return 0; // Return 0 if n is less than 1 } int sum = 0; for (int i = 1; i <= n; i += 2) { // Increment by 2 to get only odd integers sum += i; // Add each odd integer to the sum } return sum; } public static void main(String[] args) { System.out.println("Sum of odd positive integers up to 5: " + sumOfOddPositiveIntegers(5)); // 9 (1 + 3 + 5) System.out.println("Sum of odd positive integers up to 10: " + sumOfOddPositiveIntegers(10)); // 25 (1 + 3 + 5 + 7 + 9) System.out.println("Sum of odd positive integers up to 0: " + sumOfOddPositiveIntegers(0)); // 0 System.out.println("Sum of odd positive integers up to -3: " + sumOfOddPositiveIntegers(-3)); // 0 } } </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 SquareSumCalculator { public static int sumOfSquares(int n) { if (n < 1) { return 0; // Return 0 if n is less than 1 } int sum = 0; for (int i = 1; i <= n; i++) { sum += i * i; // Add the square of each integer to the sum } return sum; } public static void main(String[] args) { System.out.println("Sum of squares of positive integers up to 5: " + sumOfSquares(5)); // 55 (1^2 + 2^2 + 3^2 + 4^2 + 5^2) System.out.println("Sum of squares of positive integers up to 10: " + sumOfSquares(10)); // 385 (1^2 + 2^2 + ... + 10^2) System.out.println("Sum of squares of positive integers up to 0: " + sumOfSquares(0)); // 0 System.out.println("Sum of squares of positive integers up to -3: " + sumOfSquares(-3)); // 0 } } </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 VowelCounter { public static int countVowels(String input) { if (input == null) { return 0; // Return 0 if the input string is null } } } </pre>

		<pre> } int count = 0; String lowerCaseInput = input.toLowerCase(); for (char ch : lowerCaseInput.toCharArray()) { if (ch == 'a' ch == 'e' ch == 'i' ch == 'o' ch == 'u') { count++; // Increment the count for each vowel found } } return count; // Return the total count of vowels } public static void main(String[] args) { System.out.println("Number of vowels in 'Hello World': " + countVowels("Hello World")); // 3 System.out.println("Number of vowels in 'Java Programming': " + countVowels("Java Programming")); // 5 System.out.println("Number of vowels in 'OpenAI': " + countVowels("OpenAI")); // 3 System.out.println("Number of vowels in ': " + countVowels("")); // 0 System.out.println("Number of vowels in null: " + countVowels(null)); // 0 } } </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 PunctuationRemover { public static String removePunctuation(String s) { if (s == null) { return null; // Return null if the input string is null </pre>


```

    }

    StringBuilder result = new StringBuilder();

    for (char ch : s.toCharArray()) {
        if (Character.isLetterOrDigit(ch) ||
Character.isWhitespace(ch)) {
            result.append(ch); // Append valid characters to the
StringBuilder
        }
    }

    return result.toString(); // Convert StringBuilder to String
and return
}

public static void main(String[] args) {
    System.out.println("Original: 'Let's try, Mike!'");
    System.out.println("Without punctuation: " +
removePunctuation("Let's try, Mike!") + ""); // "Lets try Mike"

    System.out.println("Original: 'Hello, world! How are
you?'");
    System.out.println("Without punctuation: " +
removePunctuation("Hello, world! How are you?") + ""); //
"Hello world How are you"

    System.out.println("Original: 'Java: A programming
language.'");
    System.out.println("Without punctuation: " +
removePunctuation("Java: A programming language.") + ""); //
"Java A programming language"

    System.out.println("Original: '');
    System.out.println("Without punctuation: " +
removePunctuation('') + ""); // ''

    System.out.println("Original: null");
    System.out.println("Without punctuation: " +
removePunctuation(null) + ""); // null

```

		<pre> } } </pre>
10	R-1.10	<p>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.</p> <pre> 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; } } </pre>

		<pre> public void setNumberOfPetals(int numberOfPetals) { this.numberOfPetals = numberOfPetals; } public float getPrice() { return price; } public void setPrice(float price) { this.price = price; } public static void main(String[] args) { Flower rose = new Flower("Rose", 32, 2.50f); System.out.println("Flower Name: " + rose.getName()); System.out.println("Number of Petals: " + rose.getNumberOfPetals()); System.out.println("Price: \$" + rose.getPrice()); rose.setName("Tulip"); rose.setNumberOfPetals(15); rose.setPrice(3.00f); System.out.println("\nUpdated Flower Name: " + rose.getName()); System.out.println("Updated Number of Petals: " + rose.getNumberOfPetals()); System.out.println("Updated Price: \$" + rose.getPrice()); } } </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 String cardNumber; private String cardHolderName; private double creditLimit; </pre>

		<pre> private double balance; public CreditCard(String cardNumber, String cardHolderName, double creditLimit) { this.cardNumber = cardNumber; this.cardHolderName = cardHolderName; this.creditLimit = creditLimit; this.balance = 0.0; // Initial balance is set to 0 } public boolean makePurchase(double amount) { if (amount + balance <= creditLimit) { balance += amount; return true; // Purchase successful } else { return false; // Purchase exceeds credit limit } } public void makePayment(double amount) { balance -= amount; // Deduct payment from the balance } public void updateCreditLimit(double newLimit) { if (newLimit >= balance) { // Ensure new limit is not less than current balance this.creditLimit = newLimit; } else { System.out.println("New credit limit cannot be less than the current balance."); } } public double getCreditLimit() { return creditLimit; } public double getBalance() { return balance; } </pre>
--	--	---

		<pre> public static void main(String[] args) { CreditCard myCard = new CreditCard("1234-5678-9012-3456", "John Doe", 5000.0); System.out.println("Initial Credit Limit: \$" + myCard.getCreditLimit()); System.out.println("Initial Balance: \$" + myCard.getBalance()); myCard.makePurchase(1500.0); System.out.println("Balance after purchase: \$" + myCard.getBalance()); myCard.updateCreditLimit(6000.0); System.out.println("Updated Credit Limit: \$" + myCard.getCreditLimit()); myCard.updateCreditLimit(1000.0); // Should print an error message myCard.makePayment(500.0); System.out.println("Balance after payment: \$" + myCard.getBalance()); } </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 { private String cardNumber; private String cardHolderName; private double creditLimit; private double balance; </pre>

```
public CreditCard(String cardNumber, String
cardHolderName, double creditLimit) {
    this.cardNumber = cardNumber;
    this.cardHolderName = cardHolderName;
    this.creditLimit = creditLimit;
    this.balance = 0.0; // Initial balance is set to 0
}

public boolean makePurchase(double amount) {
    if (amount + balance <= creditLimit) {
        balance += amount;
        return true; // Purchase successful
    } else {
        return false; // Purchase exceeds credit limit
    }
}

public void makePayment(double amount) {
    if (amount < 0) {
        System.out.println("Payment amount cannot be negative.
Payment ignored.");
        return; // Ignore negative payments
    }
    balance -= amount; // Deduct payment from the balance
}

public void updateCreditLimit(double newLimit) {
    if (newLimit >= balance) { // Ensure new limit is not less
than current balance
        this.creditLimit = newLimit;
    } else {
        System.out.println("New credit limit cannot be less than
the current balance.");
    }
}

public double getCreditLimit() {
    return creditLimit;
}
```

```
public double getBalance() {
    return balance;
}

public static void main(String[] args) {
    CreditCard myCard = new CreditCard("1234-5678-9012-
3456", "John Doe", 5000.0);

    System.out.println("Initial Credit Limit: $" +
myCard.getCreditLimit());
    System.out.println("Initial Balance: $" +
myCard.getBalance());

    myCard.makePurchase(1500.0);
    System.out.println("Balance after purchase: $" +
myCard.getBalance());

    myCard.makePayment(-500.0); // Should print a message
and ignore the payment

    myCard.makePayment(500.0);
    System.out.println("Balance after valid payment: $" +
myCard.getBalance());

    myCard.updateCreditLimit(6000.0);
    System.out.println("Updated Credit Limit: $" +
myCard.getCreditLimit());

    myCard.updateCreditLimit(1000.0); // Should print an error
message
}
```

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> public class CreditCard { private String cardNumber; private String cardHolderName; private double creditLimit; private double balance; public CreditCard(String cardNumber, String cardHolderName, double creditLimit) { this.cardNumber = cardNumber; this.cardHolderName = cardHolderName; this.creditLimit = creditLimit; this.balance = 0.0; } public boolean makePurchase(double amount) { if (amount + balance <= creditLimit) { balance += amount; return true; // Purchase successful } else { System.out.println(cardHolderName + "'s purchase of \$" + amount + " exceeds the credit limit."); return false; // Purchase exceeds credit limit } } } public static void main(String[] args) { // Create three CreditCard objects CreditCard card1 = new CreditCard("1234-5678-9012-3456", "Alice", 1000.0); CreditCard card2 = new CreditCard("2345-6789-0123-4567", "Bob", 1500.0); </pre>
----	--------	---

	<pre>CreditCard card3 = new CreditCard("3456-7890-1234-5678", "Charlie", 2000.0); CreditCard[] cards = {card1, card2, card3}; double[] charges = {500.0, 1200.0, 2500.0}; // card2 will exceed its limit for (int i = 0; i < cards.length; i++) { System.out.println("Attempting to charge \$" + charges[i] + " to " + cards[i].cardHolderName); cards[i].makePurchase(charges[i]); } }</pre> <p>[In this setup, card3 (Charlie) will attempt to exceed its credit limit when charging \$2500.0, which is more than its \$2000 limit. The console will output a message indicating that the purchase exceeds the credit limit.]</p>
--	---