Review of Java programming basics

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

|  |  |
| --- | --- |
| 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 AllBaseTypes { private static void checkAllBaseTypes() {  Scanner input = new Scanner(System.in);  System.out.println("Enter a byte value: ");  while (!input.hasNextByte()) {  System.out.println("please enter valid byte value ");  input.next(); }  System.out.println("you entered byte value " + input.nextByte() + " \n"); System.out.println("please enter short value: ");  while (!input.hasNextShort()) {  System.out.print("please enter valid short value: \n");  input.next(); }  System.out.println("entered short is " + input.nextShort());  input.next();  System.out.print("Enter an integer : ");  while (!input.hasNextInt()) {  input.next();  System.out.println("Please enter valid integer : "); } System.out.println("Entered Integer value is : " + input.nextInt()); System.out.print("Enter a float : ");  while (!input.hasNextFloat()) {  input.next();  System.out.println("Please enter valid float : "); } System.out.println("Entered float value is " + input.nextFloat()); System.out.print("Enter a long: ");  while (!input.hasNextLong()) {  input.next();  System.out.println("Please enter valid long vzlue: "); } System.out.println("Entered long value is : " + input.nextLong()); System.out.print("Enter a double: ");  while (!input.hasNextDouble()) {  input.next();  System.out.println("Please enter valid double value : "); } System.out.println("Entered double value is : " + input.nextDouble()); System.out.print("Enter a boolean: ");  while (!input.hasNextBoolean()) {  input.next();  System.out.println("Please enter boolean value: "); } System.out.println("Entered boolean value is : "+input.nextBoolean()); System.out.print("Enter a character ");  char c = input.next().charAt(0);  System.out.println("Entered character is : " + c);  input.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]?  classTestarray2{ public static void main(String args[]) { inA[]={33,3,4,450,120}; A[4]=550; intB[]=A.clone(); for(inti=0;i<A.length;i++)  System.out.println(A[i]); for(int  i=0;i<B.length;i++)  System.out.println(B[i]); } }  The game entry object referenced by B[4] will updated with 550 |
| 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.  import java.util.Scanner;  public class CheckMultiple {  public static void main(String[] args) {  isMultiple();  }  private static void isMultiple() throws ArithmeticException{  Scanner input = new Scanner(System.in);  int reminder;  int value1 = input.nextInt();  int value2 = input.nextInt();  reminder = value1 % value2; if (value2 == 0 || value1 == 0 || reminder != 0) {  System.out.println("entered values either zero;; OR " + value1 + " not multiple of " + value2); }  else {  System.out.println(value1 + " is Multiple of " + value2); }  input.close(); }  } |
| 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.  import java.util.Scanner;  public class EvenTest {  public static boolean isEven(long value) {  boolean isItEven = false;  for (int i = 1; i < value; i++)  isItEven = !isItEven; return isItEven; }  public static void main(String args[]) {  Scanner input = new Scanner(System.in);  System.out.print("Enter a number:");  Long userInput = input.nextLong();  boolean check = isEven(userInput);  if (check == true) {  System.out.println(+userInput + " is even number"); }  else {  System.out.println(+userInput + " is not an even number"); }  input.close();  }  } |
| 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 static int sum(int n){  return n\*(n+1)/2;  } |
|  |  |
|  |  |
| 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.  import java.util.Scanner;  public class SumOfOddPositiveIntegers {   static int number;  int oddNumbers[];    public static void main(String[] args) {  SumOfOddPositiveIntegers findSum = new SumOfOddPositiveIntegers();  findSum.getTheNumber();  findSum.findOddNumbers(number);  System.out.println("the odd numbers are");  findSum.printOddNumbers();  System.out.println();  System.out.println("the sum of all odd positive integers less than or equal to n is:  " + findSum.findSum());   }    public void getTheNumber(){  Scanner scanner = new Scanner(System.in);  System.out.println("enter the number");  number = scanner.nextInt();  oddNumbers = new int[number/2  + 1];  scanner.close();  }    public int findSum(){  int sum = 0;  for (int i = 0; i < oddNumbers.length; i++) {  sum = sum + oddNumbers[i];  }  return sum;  }    public void findOddNumbers(int n){  int j = 0;  for (int i = 1; i <= n; i++) {  if((i % 2) != 0){  oddNumbers[j++] = i;  }  }  }    public void printOddNumbers(){  for (int i = 0; i < oddNumbers.length; i++) {  System.out.print(oddNumbers[i] + " ");  }  } |
| 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.  import java.util.Scanner;  public class SumOfSquares {   static int number;    public static void main(String[] args) {  SumOfSquares findSum = new SumOfSquares();  findSum.getTheNumber();  System.out.println("the sum of the squares of all positive integers less than or equal to n is:  " + findSum.findSum(number));   }    public void getTheNumber(){  Scanner scanner = new Scanner(System.in);  System.out.println("enter the number");  number = scanner.nextInt();  scanner.close();  }    public int findSum(int number){  int sum = 0;  for (int i = 1; i <= number; i++) {  sum = sum + (i \* i);  }  return sum;  }  } |
| R-1.8 | Write a short Java method that counts the number of vowels in a given character string.  import java.io.\*;  public class vowel {  public static void main(String[] args)  throws IOException  {  String str = "GeeksForGeeks";  str = str.toLowerCase();  int count = 0;  for (int i = 0; i < str.length(); i++) {  // check if char[i] is vowel  if (str.charAt(i) == 'a' || str.charAt(i) == 'e'  || str.charAt(i) == 'i'  || str.charAt(i) == 'o'  || str.charAt(i) == 'u') {  // count increments if there is vowel in  // char[i]  count++;  }  }  // display total count of vowels in string  System.out.println(  "Total no of vowels in string are: " + 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".  import java.util.Scanner;  public class RemovePunctuationFromAString {   String sentence = "";    public static void main(String[] args) {  RemovePunctuationFromAString removePunctuationFromAString = new RemovePunctuationFromAString();  removePunctuationFromAString.getTheNumber();  System.out.println("After removing all the punctuations: " + removePunctuationFromAString.removePunctuation());   }    public void getTheNumber(){  Scanner scanner = new Scanner(System.in);  System.out.println("enter a sentence with punctuation");  sentence = scanner.nextLine();  scanner.close();  }    public String removePunctuation(){  StringBuilder outputStr = new StringBuilder();  char[] characters = sentence.toCharArray();  int asciiValue = 0;  for (int i = 0; i < characters.length; i++) {  asciiValue = (int)characters[i];  if(!(asciiValue > 32 && asciiValue < 48)){  outputStr.append(characters[i]);  }  }    return outputStr.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 {  String flower; int petals;  float price;  public Flower(String flower, int petals, float price) {  this.flower = flower; this.petals = petals;  this.price = price; }  public String getFlower() {  return flower; }  public void setFlower(String flower) {  this.flower = flower; }    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; }  public static void main(String[] args) {  Flower flower = new Flower("astor flower", 12, 6.99f);  System.out.println("the flower name, petals count and price respectively is: " + flower.getFlower() + ", " + flower.getPetals() + ", " + 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 CreditCardN {   private String customer;  private String bank;  private String account;   private int limit;  protected double balance;   public CreditCardN(String cust, String bk, String acnt, int lim, double initialBal) {   customer = cust;   bank = bk;   account = acnt;   limit = lim;   balance = initialBal;  }  public CreditCardN(String cust, String bk, String acnt, int lim) {  this(cust, bk, acnt, lim, 0.0);  }     public String getCustomer() {  return customer;     }   public String getBank() {         return bank;     }      public String getAccount() {          return account;     }   public int getLimit() {  return limit;     }  public double getBalance() {  return balance;     }   public boolean charge(double price) {  if (price + balance > limit)  return false;   balance += price;   return true;  }      public void makePayment(double amount) {   if (amount < 0) {        System.out.println("Your payment can not be negative :( ");         } else  balance -= amount;  }  public static void printSummary(CreditCardN card) System.out.println("Customer+card.customer);  System.out.println("Bank = " + card.bank);  System.out.println("Account = " + card.account);  System.out.println("Balance = " + card.balance);  System.out.println("Limit = " + card.limit); }    public static void main(String[] args) {          CreditCardN[] wallet = new CreditCardN[3];          wallet[0] = new CreditCardN("John Bowman", "California Savings", "5391 0375 9387 5309", 5000);          wallet[1] = new CreditCardN("John Bowman", "California Federal", "3485 0399 3395 1954", 3500);          wallet[2] = new CreditCardN("John Bowman", "California Finance", "5391 0375 9387 5309", 2500, 300);          for (int val = 1; val <= 16; val++) {  wallet[0].charge(3 \* val);             wallet[1].charge(2 \* val);  wallet[2].charge(val);  }  for (CreditCardN card : wallet) {  CreditCardN.printSummary(card);  if (card.getBalance() > 200.0) {     card.makePayment(-200);   System.out.println("New balance = " + card.getBalance());              }     }     }  } |
| R-1.12 | Modify the CreditCard class from Code Fragment 1.5 so that it ignores any request to process a negative payment amount. |
| 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? |