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 | 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.  //كتابة دالة قصيرة  package java\_problems\_datastructures;  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();  }  public static void main(String args[]) {  checkAllBaseTypes();}} |
| 2 | 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]?  //كم بيكون المصفوفة في[4]  class Testarray2{  public static void main(String args[])  {  int A[]={33,3,4,450,120};  A[4]=550;  int B[] = A.clone();  for(int i=0;i<A.length;i++)  System.out.println(A[i]);  for(int i=0;i<B.length;i++)  System.out.println(B[i]);  }  } |
| 3 | 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.  ////قابل للقسمة  مضاعف العدد  package java\_problems\_datastructures;  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();  }  } |
| 4 | 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.  //الرقم زوجي وتعيد قيمة true  package java\_problems\_datastructures;  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();}} |
| 5 | 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.  //تعيد جمع الاعداد الصحيحة الموجبة اقل منn  package java\_problems\_datastructures;  import java.util.Scanner;  public class SumOfIntegers {  public static void main(String args[]) {  Scanner input = new Scanner(System.in);  System.out.println("please enter a value ");  int n = input.nextInt();  System.out.println("sum of all positive integers is: " + sumOfIntegers(n));  input.close();}    private static int sumOfIntegers(int n) {    int sum = n \* (n + 1) / 2;  return sum;}} |
| 6 | 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.  //جمع الاعداد الفردية الاقل n  package java\_problems\_datastructures;  import java.util.Scanner;  public class SumOfOddNumbers {  public static void main(String args[]) {  Scanner scanner = new Scanner(System.in);  System.out.println("enter the number \n");  int number = scanner.nextInt();  }}  int[] oddNumbers = new int[number / 2 + 1];  scanner.close();  int i = 0;  long sum = 0;  for (int odd = 1; odd <= number; odd++) {  if (odd % 2 == 1) {  oddNumbers[i++] = odd;    for (i = 0; i < oddNumbers.length; i++) {  sum += oddNumbers[i];}  System.out.println("sum of all odd positive integers less than or equal to " + number + " is " + sum);}} |
| 7 | 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.  جمع مربع الاعداد الصحيحة والاقل من n //  package java\_problems\_datastructures;  import java.util.Scanner;  public class SumOfSquares {  static int number;  long sum;  private void takeInput() {  Scanner scanner = new Scanner(System.in);  System.out.println("please enter a number ");  number = scanner.nextInt();  scanner.close();  }  private long sumOfSquareIntegers(int n) {  for (int i = 0; i <= number; i++) {  sum += (i \* i);  }  return sum;}  public static void main(String[] args) {  SumOfSquares squares = new SumOfSquares();  squares.takeInput();  System.out.println("sum of squares of all positive integers less than or equal to " + number +" is: "+ squares.sumOfSquareIntegers(number));}} |
| 8 | R-1.8 | Write a short Java method that counts the number of vowels in a given character string.  // حسب كم عدد الاحرف الساكنة  package java\_problems\_datastructures;  import java.util.Scanner;  public class CountVowels {  static String string;  private static void enterString() {  Scanner input = new Scanner(System.in);  System.out.println("Enter a sring value: ");  string = input.nextLine();    string.toLowerCase();  string = string.replaceAll("\\s", "");  input.close();}  private static int countVowels(String s) {  int count = 0;  char[] charArray = string.toCharArray();  for (int i = 0; i < string.length(); i++) {  if (charArray[i] == 'a' || charArray[i] == 'e' || charArray[i] == 'i' || charArray[i] == 'o'  || charArray[i] == 'u') {  count++;}}  return count;}  public static void main(String[] args) {  enterString();  System.out.println("Total vowels in the string is " + countVowels(string));}} |
| 9 | 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".  //يبعد الرموز والفواصل  package java\_problems\_datastructures;  import java.util.Scanner;  public class CountVowels {  static String str;  private static void enterString() {  Scanner input = new Scanner(System.in);  System.out.println("Enter a sring value: ");  str = input.nextLine();  input.close();}  private static String removePunctuations(String string) {  string = string.replaceAll("\\p{P}", "");  return string;}  public static void main(String[] args) {  enterString();  System.out.println("After removing the all punctuations in the string is: " + removePunctuations(str));}} |
| 10 | 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.  //انشاء كلاس الفلور  package java\_problems\_datastructures;  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() + ", ");}} |
| 11 | R-1.11 | Modify the CreditCard class from Code Fragment 1.5 to include a method that updates the credit limit.  // تعديل الدالة  public class CreditCard {  private String customer;   private String bank;   private String account;   private double balance;   private double creditLimit;   public CreditCard(String cust, String bk, String acnt, double limit) {  customer = cust;  bank = bk;  account = acnt;  creditLimit = limit;  balance = 0.0; }  public String getCustomer() { return customer; }  public String getBank() { return bank; }  public String getAccount() { return account; }  public double getBalance() { return balance; }  public double getCreditLimit() { return creditLimit; }  public boolean charge(double price) {  if (price + balance > creditLimit) {  return false;   }  balance += price;  return true; }  public void makePayment(double amount) {  if (amount > 0) {  balance -= amount;}}  public boolean setCreditLimit(double newLimit) {  if (newLimit < balance) {  return false;}  creditLimit = newLimit;  return true; }  public static void printSummary(CreditCard 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("Credit Limit = " + card.creditLimit); }} |
| 12 | R-1.12 | Modify the CreditCard class from Code Fragment 1.5 so that it ignores any request to process a negative payment amount.  //يعدل فروم الكرد ولتحديث الائتمانmian  package java\_problems\_datastructures;  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); // calling static method  if (card.getBalance() > 200.0) {  card.makePayment(-200); // calling payment method passing negative value as payment.  System.out.println("New balance = " + card.getBalance());}}}} |
| 13 | 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?  //تجاهل اي طلب للمعالجة  public static void main(String[] args) {  CreditCard[] wallet = new CreditCard[3];  wallet[0] = new CreditCard("John Bowman", "California Savings", "5391 0375 9387 5309", 1000);  wallet[1] = new CreditCard("John Bowman", "California Federal", "3485 0399 3395 1954", 2000);  wallet[2] = new CreditCard("John Bowman", "California Finance", "6011 4902 3294 2994", 3000);  for (int j = 1; j <= 16; j++) {  wallet[0].charge(50);  wallet[1].charge(150 );  wallet[2].charge(75); } |