* ***Hello world program***

**package labr1.pkg1;**

**public class LabR11 {**

**public static void main(String[] args) {**

**System.out.println("Hello world");**

**}**

**}**

* ***Print numbers from 1-10***

**package labr1.pkg1a;**

**public class LabR11a {**

**public static void main(String[] args) {**

**for(int i=1;i<=10;i++){**

**System.out.println(i);**

**}**

**}**

**}**

* ***Print array elements***

**package labr1.pkg1b;**

**public class LabR11b {**

**public static void main(String[] args) {**

**int[] kim={12,5,3,9};**

**for(int i=0;i<kim.length;i++){**

**System.out.println(kim[i]);**

}}}

* ***Input array elements***

**package lab1.pkg1c;**

**import java.util.Scanner;**

**public class Lab11c {**

**public static void main(String[] args) {**

**Scanner kim=new Scanner(System.in);**

**System.out.println("enter the number");**

**int[]jk=new int[4];**

**for(int v=0;v<jk.length;v++){**

**jk[v]=kim.nextInt();**

**System.out.println(jk[v]);**

**}**

**}} }}**

* ***Define method to print array elements***

**package lab1.pkg1d;**

**public class Lab11d {private**

**int[] rm={1,2,3,4};**

**public void arrylist(){**

**for(int i=0;i<rm.length;i++){**

**System.out.println(rm[i]);**

**}}**

**public static void main(String[] args) {**

**Lab11d u=new Lab11d();**

**u.arrylist();**

**}**

**}**

* ***Define method to Input array elements***

**package lab1.pkg1e;**

**import java.util.Scanner;**

**public class Lab11e {**

**int[]rm=new int[4];**

**Scanner m=new Scanner(System.in);**

**public void arrylist(){**

**for(int i=0;i<rm.length;i++){**

**rm[i]=m.nextInt();**

**System.out.println(rm[i]);**

**}}**

**public static void main(String[] args) {**

**System.out.println("enter the number");**

**Lab11e w=new Lab11e();**

**w.arrylist();**

**}**

**}**

* ***Array of objects (Students)***

**package lab.pkg1.pkg1f;**

**class STudent{String name;**

**int age;int id;**

**public STudent(String name, int age,int id) {**

**this.name = name;**

**this.age = age;**

**this.id = id;**

**}**

**public int getId() {**

**return id;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**public void setAge(int age) {**

**this.age = age;**

**}**

**public String getName() {**

**return name;**

**}**

**public int getAge() {**

**return age;**

**}**

**public void display() {**

**System.out.println("Name: " + name );**

**System.out.println("ID: " + id );**

**System.out.println("Age: " + age );**

**}**

**}**

**public class Lab11f {**

**public static void main(String[] args) {**

**// TODO code application logic here**

**STudent[] plet=new STudent[1];**

**plet[0] = new STudent("kimtaehyung", 27, 1998);**

**for (STudent student : plet) {**

**student.display();**

**}**

**}**

**} }**

**}**

* ***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.***

**package lab1.pkg1g;**

**import java.util.Scanner;**

**public class Lab11g {**

**public void InputAllbasetypel(){**

**Scanner w=new Scanner(System.in);**

**System.out.print("enter the name: ");**

**String b=w.nextLine();**

**System.out.print("enter the age: ");**

**int f=w.nextInt();**

**System.out.print("Enter a byte value: ");**

**byte q = w.nextByte();**

**System.out.print("Enter a long value: ");**

**long a = w.nextLong();**

**System.out.print("Enter a double value: ");**

**double y = w.nextDouble();**

**System.out.print("Enter a float value: ");**

**float t = w.nextFloat();**

**System.out.print("Enter a char value: ");**

**char o = w.next().charAt(0);**

**System.out.print("Enter a boolean value: ");**

**boolean d = w.nextBoolean();**

**System.out.print("Enter a short value: ");**

**short p = w.nextShort();**

**System.out.println("the name: "+b);**

**System.out.println("the age: "+f);**

**System.out.println("a byte value: "+q);**

**System.out.println("a long value: "+a);**

**System.out.println("a double value: "+y);**

**System.out.println("a float value: "+t);**

**System.out.println("a char value: "+o);**

**System.out.println("a boolean value: "+d);**

**System.out.println("a short value: "+p);**

**}**

**public static void main(String[] args) {**

**Lab11g y=new Lab11g();**

**y.InputAllbasetypel();**

**}**

**}**

* ***-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]?***

**package lab1.pkg1h;**

**class GameEntry {**

**int score;**

**public GameEntry(int score) {**

**this.score = score;**

**}**

**public int getScore() {**

**return score;**

**}**

**// Setter for score**

**public void setScore(int score) {**

**this.score = score;**

**}**

**}**

**public class Lab11h {**

**public static void main(String[] args) {**

**GameEntry[] A = new GameEntry[5];**

**for (int i = 0; i < A.length; i++) {**

**A[i] = new GameEntry(i \* 100);**

**}**

**GameEntry[] B = A.clone();**

**A[4].setScore(550);**

**System.out.println("Score in B[4]: " + B[4].getScore());**

**}**

}

* ***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.***

**package lab1.pkg1i;**

**public class Lab11i {**

**public static boolean isMultiple(long n, long m) {**

**return m != 0 && n % m == 0;**

**}**

**public static void main(String[] args) {**

**System.out.println(isMultiple(15, 0));**

**System.out.println(isMultiple(15, 4));**

**System.out.println(isMultiple(0, 5));**

**}**

**}**

* ***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.***

**package lab1.pkg1j;**

**public class Lab11j {**

**public static boolean isEven(int i) {**

**while (i > 1) {**

**i = i - 2;**

**}**

**return i == 0;**

**}**

**public static void main(String[] args) {**

**System.out.println(isEven(4));**

**System.out.println(isEven(7));**

**System.out.println(isEven(1));**

**}**

**}**

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

**package lab1.pkg1k;**

**public class Lab11k {**

**public static int sum(int n) {**

**int sum = 0;**

**for (int i = 1; i<= n; i++) {**

**sum += i;**

**}**

**return sum;**

**}**

**public static void main(String[] args) {**

**System.out.println(sum(5));**

**System.out.println(sum(10));**

**}**

**}**

* ***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.***

**package lab1.pkg1l;**

**public class Lab11l {**

**public static int sumOfOdd(int n) {**

**int sum = 0;**

**for (int i = 1; i <= n; i += 2) {**

**sum += i;**

**}**

**return sum;**

**}**

**public static void main(String[] args) {**

**System.out.println(sumOfOdd(5));**

**System.out.println(sumOfOdd(10));**

**}**

**}**

***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.***

**package lab1.pkg1m;**

**public class Lab11m {**

**public static int sumOfSquares(int n) {**

**if (n <= 0) {**

**return 0;**

**}**

**int sum = 0;**

**for (int i = 1; i <= n; i++) {**

**sum += i \* i;**

**}**

**return sum;**

**}**

**public static void main(String[] args) {**

**int n = 7;**

**System.out.println("Sum of squares " + n + " is: " + sumOfSquares(n));**

**}**

**}**

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

**import java.util.Scanner;**

**public class Pract {**

**public static int count() {**

**int count = 0;**

**Scanner scan = new Scanner(System.in);**

**System.out.println("enert sentence");**

**String str = scan.nextLine();**

**for (int i = 0; i < str.length(); i++) {**

**char c = str.charAt(i);**

**if (c == 'a' || c == 'i' || c == 'e' || c == 'u' || c == 'o') {**

**count++;**

**}**

**}**

**return count;**

**}**

**public static void main(String[] args) {**

**System.out.println(count());**

**}**

**}**

***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".***

**package lab1.pkg1.l;**

**public class Lab11L {**

**public static String removePunctuation(String s) {**

**StringBuilder result = new StringBuilder();**

**for (int i = 0; i < s.length(); i++) {**

**char kim = s.charAt(i);**

**if (Character.isLetterOrDigit(kim) || Character.isWhitespace(kim)) {**

**result.append(kimc);**

**}**

**}**

**return result.toString();**

**}**

**public static void main(String[] args) {**

**String sentence = "Let’s try, Mike!";**

**System.out.println("Original: " + sentence);**

**System.out.println("Without punctuation: " + removePunctuation(sentence));**

**}**

**}**

***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.***

**package lab1.pkg1o;**

**public class Lab11o {**

**private String name;**

**private int petals;**

**private float price;**

**public Lab11o(String name, int petals, float price) {**

**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) {**

**if (petals >= 0) {**

**this.petals = petals;**

**} else {**

**System.out.println("Number of petals cannot be negative.");**

**}**

**}**

**public float getPrice() {**

**return price;**

**}**

**public void setPrice(float price) {**

**if (price >= 0) {**

**this.price = price;**

**} else {**

**System.out.println("Price cannot be negative.");**

**}**

**}**

**// Main method for testing**

**public static void main(String[] args) {**

**Lab11o rose = new Lab11o("kimtaehyung", 7, 8.79f);**

**System.out.println("Flower: " + rose.getName());**

**System.out.println("Petals: " + rose.getPetals());**

**System.out.println("Price: $" + rose.getPrice());**

**rose.setName("Tulip");**

**rose.setPetals(7);**

**rose.setPrice(7.71f);**

**System.out.println("\nUpdated Flower:");**

**System.out.println("Flower: " + rose.getName());**

**System.out.println("Petals: " + rose.getPetals());**

**System.out.println("Price: $" + rose.getPrice());**

**}**

**}**

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

**public class CreditCard {**

**private String customer;**

**private String bank;**

**private String account;**

**private int limit;**

**protected double balance;**

**public CreditCard(String cust, String bk, String acnt, int lim, double initialBal) {**

**customer = cust;**

**bank = bk;**

**account = acnt;**

**limit = lim;**

**balance = initialBal;**

**}**

**public CreditCard(String cust, String bk, String acnt, int lim) {**

**this(cust, bk, acnt, lim, 0.0);**

**}**

**// Accessor methods**

**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;**

**}**

**// Update methods**

**public boolean charge(double price) {**

**if (price + balance > limit) // if charge would surpass limit**

**return false; // refuse the charge**

**balance += price; // update the balance**

**return true; // announce the good news**

**}**

**public void makePayment(double amount) {**

**balance -= amount;**

**}**

**// New method to update the credit limit**

**public void updateLimit(int newLimit) {**

**if (newLimit > 0) {**

**limit = newLimit;**

**System.out.println("Credit limit updated to: " + limit);**

**} else {**

**System.out.println("Invalid credit limit. It must be positive.");**

**}**

**}**

**// Utility method to print a card's information**

**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("Limit = " + card.limit);**

**}**

**public static void main(String[] args) {**

**CreditCard[] wallet = new CreditCard[3];**

**wallet[0] = new CreditCard("John Bowman", "California Savings", "5391 0375 9387 5309", 5000);**

**wallet[1] = new CreditCard("John Bowman", "California Federal", "3485 0399 3395 1954", 3500);**

**wallet[2] = new CreditCard("John Bowman", "California Finance", "5391 0375 9387 5309", 2500, 300);**

**for (int val = 1; val <= 20; val++) {**

**wallet[0].charge(3 \* val);**

**wallet[1].charge(10 \* val);**

**wallet[2].charge(val);**

**}**

**for (CreditCard card : wallet) {**

**CreditCard.printSummary(card); // Print card information**

**while (card.getBalance() > 200.0) {**

**card.makePayment(200);**

**System.out.println("New balance = " + card.getBalance());**

**}**

**}**

***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 {**

**// Instance variables**

**private String customer;**

**private String bank;**

**private String account;**

**private int limit;**

**protected double balance;**

**// Constructors**

**public CreditCard(String cust, String bk, String acnt, int lim, double initialBal) {**

**customer = cust;**

**bank = bk;**

**account = acnt;**

**limit = lim;**

**balance = initialBal;**

**}**

**public CreditCard(String cust, String bk, String acnt, int lim) {**

**this(cust, bk, acnt, lim, 0.0);**

**}**

**// Accessor methods**

**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;**

**}**

**// Update methods**

**public boolean charge(double price) {**

**if (price + balance > limit) // if charge would surpass limit**

**return false; // refuse the charge**

**balance += price; // update the balance**

**return true;**

**}**

**public void makePayment(double amount) {**

**if (amount <= 0) { // Ignore negative or zero payment**

**System.out.println("Invalid payment amount: " + amount + ". Payment must be positive.");**

**Return ;**

**}**

**balance -= amount;**

**System.out.println("Payment of " + amount + " processed. New balance: " + balance);**

**}**

**// Utility method to print a card's information**

**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("Limit = " + card.limit);**

**}**

**// Main method**

**public static void main(String[] args) {**

**CreditCard card = new CreditCard("John Bowman", "California Savings", "5391037593875309", 5000);**

**System.out.println("Initial Summary:");**

**printSummary(card);**

**System.out.println("\nAttempting a valid payment of $200:");**

**card.makePayment(200);**

**printSummary(card);**

**System.out.println("\nAttempting a negative payment of -$50:");**

**card.makePayment(-50); // Should print an error message**

**printSummary(card);**

***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 CreditCard {**

**// Instance variables**

**private String customer;**

**private String bank;**

**private String account;**

**private int limit;**

**protected double balance;**

**public CreditCard(String cust, String bk, String acnt, int lim, double initialBal) {**

**customer = cust;**

**bank = bk;**

**account = acnt;**

**limit = lim;**

**balance = initialBal;**

**}**

**public CreditCard(String cust, String bk, String acnt, int lim) {**

**this(cust, bk, acnt, lim, 0.0); // use a balance of zero as default**

**}**

**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) {**

**balance -= amount;**

**}**

**}**

**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("Limit = " + card.limit);**

**}**

**public static void main(String[] args) {**

**CreditCard[] wallet = new CreditCard[3];**

**wallet[0] = new CreditCard("John Bowman", "California Savings", "5391037593875309", 5000);**

**wallet[1] = new CreditCard("John Bowman", "California Federal", "3485039933951954", 3500);**

**wallet[2] = new CreditCard("John Bowman", "California Finance", "5391037593875309", 2500, 300);**

**// Adjusted loop to cause one card to exceed the limit**

**for (int val = 1; val <= 20; val++) {**

**wallet[0].charge(3 \* val);**

**wallet[1].charge(10 \* val);**

**wallet[2].charge(val);**

**}**

**for (CreditCard card : wallet) {**

**CreditCard.printSummary(card); // Print card information**

**while (card.getBalance() > 200.0) {**

**card.makePayment(200);**

**System.out.println("New balance = " + card.getBalance());**

**}**

**}**

**}}**