**Java Basic Assignment Questions**

**Q1.Write a Java program to return true if a given string contain the string 'pop', but the middle 'o' also may other character.**

**Ans:**

*public class* Main {  
 *public static void* main(String[] args){  
 String str = "pop";  
 *for* (*int* i=0; i<str.length()-2; i++){  
 *if*( str.charAt(i) == 'p' && str.charAt(i+2) == 'p'){  
 System.out.println("True");  
 }  
 *else*  
System.out.println("False");  
 }  
 }  
}

**Q2.Write a Java program to sort a given binary array in linear times.**

**Linear time: An algorithm is said to take linear time, or O(n) time, if its time complexity is O(n). Informally, this means that the running time increases at most linearly with the size of the input. More precisely, this means that there is a constant c such that the running time is at most cn for every input of size n. For example, a procedure that adds up all elements of a list requires time proportional to the length of the list, if the adding time is constant, or, at least, bounded by a constant.**

**Example:**

**Input :**

**b\_nums[] = { 0, 1, 1, 0, 1, 1, 0, 1, 0, 0 }**

**Output:**

**After sorting: [0, 0, 0, 0, 0, 1, 1, 1, 1, 1]**

**Ans:**

*public class* Main {  
 *public static void* main(String[] args) {  
 *int* count = 0;  
 *int* flag = 1;  
 *int* b\_nums[] = {0, 1, 1, 0, 1, 1, 0, 1, 0, 0};  
 *for* (*int* i = 0; i < b\_nums.length; i++) {  
 *if* (b\_nums[i] == 0) {  
 count++;  
 }  
 *else if*(b\_nums[i] == 1){  
 flag++;  
 }  
 }  
 *for*(*int* i=0; i<count; i++){  
 b\_nums[i] = 0;  
 }  
 *for*(*int* i=count; i<b\_nums.length; i++){  
 b\_nums[i] = 1;  
 }  
 *for*(*int* val: b\_nums){  
 System.out.println(val+" ");  
 }  
 }  
}

**Q3. Write a Java program to remove a specified character from a given string.**

**Sample Output:**

**Original string: abcdefabcdeabcdaaa**

**Second string: bcdefbcdebcd**

**Ans.**

*public class* Main {  
 *public static void* main(String[] args){  
 String originalString = "abcdefabcdeabcdaaa";  
 *char* removeChar = 'a';  
 StringBuilder SecondString = *new* StringBuilder("");  
 *for* (*int* i=0; i<= originalString.length()-1; i++){  
 *if*( removeChar != originalString.charAt(i)){  
 SecondString.append(originalString.charAt(i));  
 }  
 }  
 System.out.println(SecondString);  
 }  
}

**Q4.Write a Java Program to reverse a string without using String inbuilt function.**

**Ans:**

*public class* Main {  
 *public static void* main(String[] args){  
 String s1 = "INDIAN";  
 StringBuilder s2 = *new* StringBuilder("");  
 *for*(*int* i=s1.length()-1; i>=0; i--){  
 s2.append(s1.charAt(i));  
 }  
 System.out.println(s2);  
 }  
}

### **Q5.****First Factorial**

**Have the function FirstFactorial(num) take the num parameter being passed and return the factorial of it. For example: if num = 4, then your program should return (4 \* 3 \* 2 \* 1) = 24. For the test cases, the range will be between 1 and 18 and the input will always be an integer.**

#### **Examples**

**Input: 4**

**Output: 24**

**Input: 8**

**Output: 40320**

**Ans:**

*public class* Factorial {  
 *int* fact = 1;  
 *public void* firstFactorial(*int* num){  
 *for*(*int* i=1; i<=num; i++){  
 fact = fact \* i;  
 }  
 }  
 *public static void* main(String[] args){  
 Factorial f1 = *new* Factorial();  
 f1.firstFactorial(8);  
 System.out.println(f1.fact);  
 }  
}

**Q6. Write a program to find the number of occurrences of the duplicate words in a string and print them?**

**Ans:**

*import* java.util.HashMap;  
*import* java.util.Map;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 Map<String,Integer> map=*new* HashMap<String,Integer>();  
 String str = "Checking for duplicate sentence. Checking for duplicate sentence";  
 String[] array = str.split("\\s");  
 *for*(*int* i=0; i<array.length; i++){  
 *if*(!map.containsKey(array[i]))  
 map.put(array[i],1);  
 *else*  
map.put(array[i],map.get(array[i])+1);  
 }  
 System.out.println(map);  
 }  
  
  
}

**Q7. Write a program to find the number of occurrences of a character in a string without using a loop?**

**Ans:**

*public class* FindOccurance {  
 *public static void* main(String[] args) {  
 String str = "HONULULU";  
 *char* ch = 'U';  
 String chStr = ch + "";  
 System.out.println("Given String --> "+str + "\n" + "To find the occuarance --> " + ch);  
 String AfterDeletion = str.replace(chStr,"");  
 System.out.println("Total number of occurance of character --> " + AfterDeletion.length());  
 }  
}

**Q8.Write a program to print your Firstname, LastName & age using static block, static method & static variable respectively**

**Ans:**

*public class* Main {  
 *// Static Variable*  
 *static* String FirstName = "Shivam";  
 *static* String LastName = "Roy";  
 *static int* Age = 24;  
  
 *// Static Block*  
  
 *static*{  
 System.out.println("Printing using static block" + "\n");  
 System.out.println("First Name -->> " + FirstName +  
 "\n" + "Last Name -->> " + LastName +  
 "\n" + "Age -->> " + Age);  
 }  
  
 *// Static Method*  
 *public static void* main(String[] args) {  
 System.out.println();  
 System.out.println("Printing using static main method" + "\n");  
 System.out.println( "First Name -->> " + FirstName + "\n" + "Last Name -->> " + LastName +  
 "\n" + "Age -->> " + Age);  
 }  
}

**Q9.Create 3 subclasses of bank SBI, BOI, ICICI all 4 should have a method called get details which provide there specific details like rate of interest etc, print details of every bank.**

**Ans:**

// Bank class

*package* Ques9;  
  
*public abstract class* Bank {  
 *abstract public void* getDetails();  
}

// ICICI class

*package* Ques9;  
  
*public class* ICICI *extends* Bank{  
 String bankName = "ICICI Bank";  
 *int* rateOfInterest = 12;  
 *@Override*  
 *public void* getDetails() {  
 System.out.println("Name of Bank -->> " + bankName + "\n" + "Rate of interest -->> " + rateOfInterest + "\n");  
 }  
}

// BOI class

*package* Ques9;  
  
*public class* BOI *extends* Bank{  
 String bankName = "Bank of India";  
 *int* rateOfInterest = 11;  
 *@Override*  
 *public void* getDetails() {  
 System.out.println("Name of Bank -->> " + bankName + "\n" + "Rate of interest -->> " + rateOfInterest + "\n");  
 }  
}

// SBI class

*package* Ques9;  
  
*public class* SBI *extends* Bank{  
 String bankName = "Central Bank of India";  
 *int* rateOfInterest = 12;  
 *@Override*  
 *public void* getDetails() {  
 System.out.println("Name of Bank -->> " + bankName + "\n" + "Rate of interest -->> " + rateOfInterest + "\n");  
 }  
}

// Main method

*package* Ques9;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 Bank bank = *new* BOI();  
 bank.getDetails();  
  
 bank = *new* ICICI();  
 bank.getDetails();  
  
 bank = *new* SBI();  
 bank.getDetails();  
  
 }  
}

**Q10. WAP to read words from the keyboard until the word done is entered. For each word except done, report whether its first character is equal to its last character. For the required loop, use a**

1. **while statement**
2. **do-while statement**

**Ans:**

*import* java.util.Scanner;  
  
*public class* Main {  
 *static void* whileLoopTest(){  
 System.out.println("Checking while loop : ");  
 Scanner sc = *new* Scanner(System.in);  
 String str = sc.next();  
 *while* (!str.equalsIgnoreCase("done")){  
 *if* (str.charAt(0) == str.charAt(str.length()-1))  
 System.out.println("Character are equal");  
 *else*  
System.out.println("Character are not equal");  
 str = sc.next();  
 }  
 }  
  
*static void* doWhileLoopTest()  
{  
 System.out.println("Checking do-while : ");  
 Scanner sc = *new* Scanner(System.in);  
 String str = sc.next();  
 *if*(str.equalsIgnoreCase("done"))  
 *return*;  
  
 *do*{  
 *if*(str.charAt(0 )== str.charAt(str.length()-1))  
 System.out.println("Character are Equal");  
 *else*  
System.out.println("Character are Not Equal");  
 str = sc.next();  
 }  
 *while*(!str.equalsIgnoreCase("done"));  
  
}  
  
  
  
 *public static void* main(String[] args) {  
 whileLoopTest();  
 doWhileLoopTest();  
 }  
}

**Q11. Design classes having attributes and methods (only skeleton) for a coffee shop. There are three different actors in our scenario and I have listed the different actions they do also below**

**\* Customer**

**- Pays the cash to the cashier and places his order, get a token number back**

**- Waits for the intimation that order for his token is ready**

**- Upon intimation/notification he collects the coffee and enjoys his drink**

**( Assumption: Customer waits till the coffee is done, he won’t timeout and cancel the order. The customer always likes the drink served. Exceptions like he not liking his coffee, he got wrong coffee are not considered to keep the design simple.)**

**\* Cashier**

**- Takes order and payment from the customer**

**- Upon payment, creates an order and places it into the order queue**

**- Intimates the customer that he has to wait for his token and gives him his token**

**( Assumption: Token returned to the customer is the order id. Order queue is unlimited. With a simple modification, we can design for a limited queue size)**

**\* Barista**

**- Gets the next order from the queue**

**- Prepares the coffee**

**- Places the coffee in the completed order queue**

**- Places a notification that order for token is ready**

**Ans:**

*package* Ques11;  
  
*i**mport* java.util.Objects;  
*import* java.util.Scanner;  
  
*public class* CoffeeShop {  
 *static* String order;  
 *static int* cost;  
 *static int* orderID;  
  
 *public void* Cashier(){  
 System.out.println("We are having these coffees : ");  
 System.out.println("Espresso Coffee - Rs. 50 /- " + "\n" +  
 "Americano Coffee - Rs. 80 /- " + "\n" +  
 "Cappuccino - Rs. 120 /- " + "\n" +  
 "Tea in stock - Rs. 30 /- " + "\n" +  
 "Flat White - Rs. 60 /- " + "\n" +  
 " Caffe Mocha - Rs. 150 /- ");  
 System.out.println("Enter you order : ");  
 Scanner sc = *new* Scanner(System.in);  
 order = sc.nextLine();  
 System.out.println("Please pay the total bill : ");  
 cost = sc.nextInt();  
 }  
  
 *public void* Customer(){  
 *if* (Objects.equals(order,"Espresso Coffee") && cost == 50){  
 orderID = 1;  
 System.out.println("Your token Number = " + orderID);  
 System.out.println("Wait for few minutes while your order is preparing");  
 }  
 *else if*(Objects.equals(order,"Americano Coffee") && cost == 80 ){  
 orderID = 2;  
 System.out.println("Your token Number = " + orderID);  
 System.out.println("Wait for few minutes while your order is preparing");  
 }  
 *else if* (Objects.equals(order,"Cappuccino") && cost == 120 ){  
 orderID = 3;  
 System.out.println("Your toke Number = " + orderID);  
 System.out.println("Wait for few minutes while your order is preparing");  
 }  
 *else if*( Objects.equals(order,"Tea in Stock") && cost == 30 ){  
 orderID = 4;  
 System.out.println("Your token Number = "+ orderID);  
 System.out.println("Wait for few minutes while your order is preparing");  
 }  
 *else if*(Objects.equals(order,"Flat White") && cost == 60){  
 orderID = 5;  
 System.out.println("Your token Number = " + orderID);  
 System.out.println("Wait for few minutes while your order is preparing");  
 }  
 *else if*(Objects.equals(order,"Caffe Mocha") && cost == 150){  
 orderID = 6;  
 System.out.println("Your token Number = "+ orderID);  
 System.out.println("Wait for few minutes while your order is preparing");  
 }  
 *else*  
System.out.println("You have entered the wrong choice");  
 }  
  
 *public void* Barista(){  
 *if* (orderID == 1){  
 System.out.println("Please collect you order : ");  
 System.out.println("Order = " + order + "\n" +  
 "Total amount = "+ cost + "\n" + "Token Number = " + orderID);  
 }  
 *else if*( orderID == 2){  
 System.out.println("Please collect you order : ");  
 System.out.println("Order = " + order + "\n" +  
 "Total amount = "+ cost + "\n" + "Token Number = " + orderID);  
 }  
 *else if* ( orderID == 3){  
 System.out.println("Please collect you order : ");  
 System.out.println("Order = " + order + "\n" +  
 "Total amount = "+ cost + "\n" + "Token Number = " + orderID);  
 }  
 *else if* ( orderID == 4){  
 System.out.println("Please collect you order : ");  
 System.out.println("Order = " + order + "\n" +  
 "Total amount = "+ cost + "\n" + "Token Number = " + orderID);  
 }  
 *else if* ( orderID == 5){  
 System.out.println("Please collect you order : ");  
 System.out.println("Order = " + order + "\n" +  
 "Total amount = "+ cost + "\n" + "Token Number = " + orderID);  
 }  
 *else if* ( orderID == 6 ){  
 System.out.println("Please collect you order : ");  
 System.out.println("Order = " + order + "\n" +  
 "Total amount = "+ cost + "\n" + "Token Number = " + orderID);  
 }  
 }  
  
 *public static void* main(String[] args) {  
 CoffeeShop coffeeShop = *new* CoffeeShop();  
 coffeeShop.Cashier();  
 coffeeShop.Customer();  
 coffeeShop.Barista();  
 }  
}

**Q12. Create a custom exception that do not have any stack trace**

**Ans:**

// CustomException class

*package* Ques12;  
  
*public class* CustomException *extends* Exception{  
  
 CustomException(String message){  
 System.out.println(message);  
 }  
}

// main method

*package* Ques12;  
  
*public class* Main {  
 *public static void* main(String[] args) *throws* CustomException {  
 *try* {  
 exceptionFunc();  
 }  
 *catch*(Throwable e) {  
 StackTraceElement[] trace = e.getStackTrace();  
 System.err.println(trace[0].toString());  
 }  
  
 }  
  
 *public static void* exceptionFunc()*throws* CustomException {  
 CustomException ce = *new* CustomException("Throwing Exception without stack trace");  
 StackTraceElement[] trace = *new* StackTraceElement[] {  
 *new* StackTraceElement("ClassName","methodName","fileName",1)  
 };  
 *// sets the stack trace elements*  
ce.setStackTrace(trace);  
 *throw* ce;  
  
 }  
 }

**Q13. Create Java classes having suitable attributes for Library management system.Use OOPs concepts in your design.Also try to use interfaces and abstract classes.**

**Ans:**

*// LibraryManagement class*

*package* Ques13;  
  
*import* java.security.PrivateKey;  
*import* java.util.Scanner;  
  
*public class* LibraryManangement *implements* Interface {  
  
 String name, email, address, bookName, feedBack;  
 *int* bookID;  
 *long* phoneNum;  
 *public static final int* Max\_Book\_Issued = 5;  
 *public static final int* Max\_lending\_Days = 10;  
  
 *@Override*  
 *public void* Registration() {  
 Scanner sc = *new* Scanner(System.in);  
 System.out.println("Enter you name : ");  
 name = sc.nextLine();  
 System.out.println("Enter your email address : ");  
 email = sc.nextLine();  
 System.out.println("Enter you Phone Number : ");  
 phoneNum = sc.nextInt();  
 }  
  
 *@Override*  
 *public void* RequestBook() {  
 Scanner sc = *new* Scanner(System.in);  
 System.out.println("Enter Book name : ");  
 bookName = sc.nextLine();  
 System.out.println("Enter Book ID : ");  
 bookID = sc.nextInt();  
 *switch* (bookID){  
 *case* 1:  
 System.out.println("Book issued : " + bookName + ", with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
 *case* 2:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 3:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 4:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 5:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 6:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 7:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 8:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 9:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 10:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 11:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 12:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 13:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
  
 *case* 14:  
 System.out.println("Book issued : " + bookName + "with Book ID : " + bookID + "\n");  
 System.out.println("Return the book with in : " + Max\_lending\_Days + " days" +"\n");  
 System.out.println("After " + Max\_lending\_Days + " days you will be charge Rs. 50 /- per day ");  
 *break*;  
 }  
 }  
  
 *@Override*  
 *public void* FeedBack() {  
 Scanner sc = *new* Scanner(System.in);  
 System.out.println("Please give feedback : ");  
 feedBack = sc.next();  
 *if* ( feedBack != *null*)  
 System.out.println("THANK YOU");  
 *else*  
System.out.println("Please give your valuable feedback. ");  
 }  
  
 *@Override*  
 *public void* BookInformation() {  
 System.out.println("Maximum book can be issued : " + Max\_Book\_Issued);  
 System.out.println("Total number of books are : ");  
 System.out.println(" PORTNOY’S COMPLAINT - Book ID --> 1 ");  
 System.out.println(" The Great Gatsby - Book ID --> 2 ");  
 System.out.println(" Don Quixote - Book ID --> 3 ");  
 System.out.println(" One Hundred Years of Solitude - Book ID --> 4 ");  
 System.out.println(" Invisible Man - Book ID --> 5 ");  
 System.out.println(" One Night at Call Center - Book ID --> 6 ");  
 System.out.println(" War and Peace - Book ID --> 7 ");  
 System.out.println(" Hamlet by William Shakespeare - Book ID --> 8 ");  
 System.out.println(" The Color Purple - Book ID --> 9 ");  
 System.out.println(" Madame Bovary - Book ID --> 10 ");  
 System.out.println(" The Divine Comedy - Book ID --> 11 ");  
 System.out.println(" Lolita - Book ID --> 112 ");  
 System.out.println(" The Brothers Karamazov - Book ID --> 13 ");  
 System.out.println(" Crime and Punishment - Book ID --> 14 ");  
  
 }  
}

// Interface

*package* Ques13;  
  
*public interface* Interface {  
 *void* Registration();  
 *void* RequestBook();  
 *void* FeedBack();  
 *void* BookInformation();  
}

// main method

*package* Ques13;  
  
*import* java.util.Scanner;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 System.out.println("Press 1 to check availability : ");  
 System.out.println("Press 2 to issue book : ");  
 System.out.println("Press 3 for give feedback : ");  
 System.out.println("Press 4 to exit : ");  
 Scanner sc = *new* Scanner(System.in);  
 *int* choice = sc.nextInt();  
 *switch* (choice){  
 *case* 1:  
 LibraryManangement lm1 = *new* LibraryManangement();  
 lm1.BookInformation();  
 *break*;  
 *case* 2:  
 LibraryManangement lm2 = *new* LibraryManangement();  
 lm2.Registration();  
 lm2.RequestBook();  
 *break*;  
 *case* 3:  
 LibraryManangement lm3 = *new* LibraryManangement();  
 lm3.FeedBack();  
 *case* 4:  
 System.out.println("Thank You");  
 *break*;  
 }  
 }  
}

**Q14.WAP to produce NoClassDefFoundError and ClassNotFoundException exception.**

**Ans:**

*public class* Main {  
 *public static void* main(String[] args) {  
 *try* {  
 Class.forName("ClassNotFoundException");  
 }  
 *catch* (ClassNotFoundException ex)  
 {  
 ex.printStackTrace();  
 }  
 }  
 }

**Q15. WAP to create a singleton class.**

**Ans:**

// SingletonTest class

*package* Ques15;  
  
*public class* SingletonTest {  
 *private static* SingletonTest instance = *null*;  
  
 *private* SingletonTest(){  
 System.out.println("constructor of class SingletonTest");  
 }  
  
 *public static* SingletonTest getInstance()  
 {  
 *if*(instance==*null*)  
 instance=*new* SingletonTest();  
 *return* instance;  
 }  
}

// main method

*package* Ques15;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 SingletonTest s1 = SingletonTest.getInstance();  
 System.out.println("Instance is created");  
 }  
}

**Q16.** **WAP to show object cloning in java using cloneable and copy constructor both.**

**Ans:**

// Student class

*package* Ques16;  
  
*public class* Student *implements* Cloneable{  
 String name;  
 *int* rollNo;  
  
 *// constructor*  
Student(String name, *int* rollNo){  
 *this*.name = name;  
 *this*.rollNo = rollNo;  
 }  
 *public* Object clone()*throws* CloneNotSupportedException{  
 *return super*.clone();  
 }  
}

// main method

*package* Ques16;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 *try* {  
 Student s1 = *new* Student("Shivam", 1564);  
 Student s2 = (Student)s1.clone();  
 System.out.println("Name -->> " + s1.name + " " + "Roll Number -->> " + s1.rollNo);  
 System.out.println("Name -->> " + s2.name + " " + "Roll Number -->> " + s2.rollNo);  
 }  
 *catch* (CloneNotSupportedException c) {}  
 }  
}

**Q17.** **WAP showing try, multi-catch and finally blocks.**

**Ans:**

*package* Ques17;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
  
 *try*{  
 *int*[] A = *new int*[5];  
 A[5] = 40/0;  
 System.out.println(A[10]);  
 }  
 *catch* (ArithmeticException e){  
 System.out.println("If arithmetic exception occur");  
 }  
 *catch* (ArrayIndexOutOfBoundsException e){  
 System.out.println("If ArrayIndexOutOfBoundsException occur");  
 }  
 *catch* (Exception e){  
 System.out.println("If exception is occur");  
 }  
 *finally* {  
 System.out.println("Finally block executed");  
 }  
 }  
}

**Q19.Convert a non-negative integer num to its English words representation.**

**Input: num = 123**

**Output: "One Hundred Twenty Three"**

**Input: num = 1234567891**

**Output: "One Billion Two Hundred Thirty Four Million Five Hundred Sixty**

**Seven Thousand Eight Hundred Ninety One"**

**Ans:**

// Conversion class

*package* Ques19;  
  
*public class* Conversion {  
 *private static* String[] lessThanTwenty = *new* String[] {"", "One", "Two", "Three", "Four", "Five", "Six", "Seven",  
 "Eight", "Nine", "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen",  
 "Seventeen", "Eighteen", "Nineteen"};  
 *private static* String[] tens = *new* String[]{"", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy",  
 "Eighty", "Ninety"};  
  
 *public* String numberToWords(*int* number) {  
 *if*(number == 0) {  
 *return* "Zero";  
 }  
 *return* helper(number);  
 }  
  
 *private* String helper(*int* number) {  
 StringBuilder result = *new* StringBuilder();  
 *if*(number >= 1000000000) {  
 result.append(helper(number/1000000000)).append(" Billion ").append(helper(number%1000000000));  
 } *else if*(number >= 1000000) {  
 result.append(helper(number/1000000)).append(" Million ").append(helper(number%1000000));  
 } *else if*( number >= 1000 ) {  
 result.append(helper(number/1000)).append(" Thousand ").append(helper(number%1000));  
 } *else if*(number >= 100) {  
 result.append(helper(number/100)).append(" Hundred ").append(helper(number%100));  
 } *else if*(number >=20) {  
 result.append(tens[number/10-1]).append(" ").append(helper(number%10));  
 } *else* {  
 result.append(lessThanTwenty[number]);  
 }  
  
 *return* result.toString().trim();  
 }  
  
}

// main method

*package* Ques19;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 Conversion conversion =*new* Conversion();  
 String output = conversion.numberToWords(2132154654);  
 System.out.println(output);  
 }  
  
}

**Q20.This assignment is supposed to make you practice your skills on interfaces, classes and OOPs.**

**Story - The system is a duck simulation game. There are ducks, each having weight and number of wings. All ducks can fly and quack, but different type of ducks fly and quack differently. For instance, let us consider the following–**

**1.****Rubber ducks don’t fly and squeak.**

**2.****Mallard ducks fly fast and quack loud.**

**3.****Redhead ducks fly slow and quack mild.**

**All ducks have following common property:**

**Type of Duck – Use enum to represent this value**

**Design classes and interfaces for the above story to meet the following requirements -**

**a.Create a duck**

**b.Show details of a duck, i.e. when you call for e.g. ShowDetails() method of a duck, duck should print its traits.**

**User following concepts: class, interfaces, polymorphism etc.**

**Ans:**

// Bird class

*package* Ques20;  
  
*public class* Bird {  
 *enum* DuckType{  
 Rubber,  
 Mallard,  
 RedHead;  
  
 }  
}

// BirdInterface interface

*package* Ques20;  
  
*public interface* BirdInterface {  
 *void* ShowDetail();  
}

// Duck class

*package* Ques20;  
  
*public class* Duck *implements* BirdInterface{  
 *private double* weight;  
 *private int* numberOfWings;  
 *private* Bird.DuckType TypeOfDuck;  
  
 *public* Duck (*double* weight, *int* numberOfWings, Bird.DuckType TypeOfDuck){  
 *this*.weight = weight;  
 *this*.numberOfWings = numberOfWings;  
 *this*.TypeOfDuck = TypeOfDuck;  
 }  
  
 *@Override*  
 *public void* ShowDetail() {  
 *if* ( TypeOfDuck == Bird.DuckType.Mallard)  
 System.out.println("Mallard Duck");  
 *else if* ( TypeOfDuck == Bird.DuckType.Rubber)  
 System.out.println("Rubber Duck");  
 *else if* ( TypeOfDuck == Bird.DuckType.RedHead)  
 System.out.println("Redhead Duck");  
 System.out.println("Weight of duck : " + weight);  
 System.out.println("Total number of wings : " + numberOfWings);  
 }  
}

// MallardDuck class

*package* Ques20;  
  
*public class* MallardDuck *extends* Duck{  
  
 *public* MallardDuck(*double* weight, *int* numberOfWings, Bird.DuckType TypeOfDuck) {  
 *super*(weight, numberOfWings, TypeOfDuck);  
 }  
 *public void* ShowDetail(){  
 *super*.ShowDetail();  
 System.out.println("Mallard ducks fly fast and quack loud.");  
 }  
}

// RedheadDuck class

*package* Ques20;  
  
*public class* RedheadDuck *extends* Duck{  
  
 *public* RedheadDuck(*double* weight, *int* numberOfWings, Bird.DuckType TypeOfDuck) {  
 *super*(weight, numberOfWings, TypeOfDuck);  
 }  
 *public void* ShowDetail(){  
 *super*.ShowDetail();  
 System.out.println("Redhead ducks fly slow and quack mild.");  
 }  
}

// RubberDuck class

*package* Ques20;  
  
*public class* RubberDuck *extends* Duck{  
 *public* RubberDuck(*double* weight, *int* numberOfWings, Bird.DuckType TypeOfDuck) {  
 *super*(weight, numberOfWings, TypeOfDuck);  
 }  
 *public void* ShowDetail(){  
 *super*.ShowDetail();  
 System.out.println("Rubber ducks don’t fly and squeak.");  
 }  
}

// main method

*package* Ques20;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 BirdInterface[] duck = *new* BirdInterface[3];  
 duck[0] = *new* RubberDuck(7,2, Bird.DuckType.Rubber);  
 duck[1] = *new* MallardDuck(8,2, Bird.DuckType.Mallard);  
 duck[2] = *new* RedheadDuck(9,2, Bird.DuckType.RedHead);  
 *for* (*int* i=0; i<=3; i++){  
 duck[i].ShowDetail();  
 System.out.println();  
 }  
 }  
}

**Q21.Write a java program to maintain the list of Employees who works in the organization. Follow the below given design to develop the above program:**

**Design a class Employee with EmpId, EmpName & EmpDesignation as private**

**attributes. It contains a parameterized constructor to initialize all the data members of class. It also comprises the Getter methods for all the private fields.**

**Design a main class comprising main method. The main class comprises an array of Employee object which needs to be initialized with the command line input.**

**It displays the information of Employees from the array using loop.**

**Ans:**

// Employee class

*package* Ques21;  
  
*public class* Employee {  
 *private long* EmpId;  
 *private* String EmpName;  
 *private* String EmpDesignation;  
  
 *public* Employee(*long* EmpId, String EmpName, String EmpDesignation){  
 *this*.EmpId = EmpId;  
 *this*.EmpName = EmpName;  
 *this*.EmpDesignation = EmpDesignation;  
 }  
  
 *public long* getEmpId(){  
 *return* EmpId;  
 }  
  
 *public* String getEmpName(){  
  
 *return* EmpName;  
 }  
  
 *public* String getEmpDesignation(){  
  
 *return* EmpDesignation;  
 }  
}

// EmpDetail class

*package* Ques21;  
  
*public class* EmpDetail {  
 *private* Employee[] detail;  
 *public void* setDetail(Employee[] detail){  
 *this*.detail = detail;  
 }  
  
 *public* Employee[] getDetail(){  
 *return* detail;  
 }  
}

// main method

*package* Ques21;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 EmpDetail emp = *new* EmpDetail();  
 Employee[] obj = *new* Employee[4];  
 obj[0] = *new* Employee(1564,"John","Sr. Software Consultant");  
 obj[1] = *new* Employee(1658,"Roman", "Content Writer");  
 obj[2] = *new* Employee(1895,"Brain","Software Consultant");  
 obj[3] = *new* Employee(1647,"Harry","Java Developer");  
 emp.setDetail(obj);  
 Employee[] emp1 = emp.getDetail();  
 *for* (Employee employeeDetail: emp1){  
 System.out.println("Name --> " + employeeDetail.getEmpName() + "\n"  
 + "Employee ID --> " + employeeDetail.getEmpId() + "\n"  
 + "Designation --> " + employeeDetail.getEmpDesignation());  
 }  
 }  
}

**Q22.Check Digits. Credit cards usually have a so-called check digit. This is a single digit that I assigned when the account number is developed and has a special property. One particularly simple mechanism is to assign the last digit of the sum of all the other digits. For example, suppose we have a nine-digit account number (including the check digit). The check digit would be the sum of the eight digits. This digit could be placed any where’s in the sequence, say the**

**third digit. As a full example, suppose the eight numbers are 12345678. Their sum is 36; thus, 6 is the check digit. The account number is therefore 126345678. Write a program to read in a nine digit integer from the keyboard and check it for these rules. Write"okay" or "not okay" for the results of the test.**

**Ans:**

// CreditCard class

*package* Ques22;  
  
*public class* CreditCard {  
 *static int* sum = 0, count = 0, num, digit;  
 *void* check(*int* num){  
 *int* length = String.valueOf(num).length();  
 *if* ( length == 9 ){  
 *while* ( num > 0 ) {  
 digit = digit % 10;  
 sum = sum + digit;  
 num = num / 10;  
 }  
 System.out.println("Sum of digit of card number : " + sum);  
 *if* ( sum % 10 == 0)  
 System.out.println("OKAY");  
 *else*  
System.out.println(" NOT OKAY");  
 }  
 *else*  
System.out.println("You have exceed the digit. Try Again! ");  
 }  
}

// main method

*package* Ques22;  
  
*import* java.util.Scanner;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 Scanner sc = *new* Scanner(System.in);  
 System.out.println("Enter 9 digit number to check : ");  
 *int* num = sc.nextInt();  
 CreditCard cc = *new* CreditCard();  
 cc.check(num);  
 }  
}