1(a) How is java different from c++? Explain the working of JRE.

|  |  |  |
| --- | --- | --- |
| ****Parameters**** | ****C++**** | ****Java**** |
| ****History**** | Bjarne Stroustrup developed C++ in 1979 at Bells Lab, and it was first released in Oct 1985 | James Gosling developed Java at Sun Microsystems, and it was first released on May 23, 1995 |
| ****Programming Paradigm**** | C++ supports procedural and Object Oriented programming language | Java only supports Object Oriented programming language |
| ****Platform Dependency**** | It is a platform-dependent language and needs to compile for different platforms | Java is platform-independent; that is, we can write once and run it anywhere |
| ****Compilation & Interpretation**** | C++ language can only be compiled and cannot be interpreted | Javalanguage can be compiled and interpreted |
| ****Memory Management**** | In C++, memory management is manual, so we need to allocate or deallocate memory manually | In Java, memory management is system-controlled |
| ****Portability**** | C++ is a non-portable language | Java is a portable language |

JRE - **JRE which stands for Java Runtime Environment. The Java Runtime Environment is a software package that provides libraries, Java Virtual Machine (JVM), and other components to run applications written in the Java programming language. Here's how it works.**

**Q1(b)- What is role of methods in java ? How can we use this keyword with local variable in java ?**

**Ans - A method in Java is a block of code that, when called, performs specific actions mentioned in it. For instance, if you have written instructions to draw a circle in the method, it will do that task. You can insert values or parameters into methods, and they will only be executed when called. They are also referred to as functions. The primary uses of methods in Java are:**

**It allows code reusability (define once and use multiple times)**

**You can break a complex program into smaller chunks of code**

**It increases code readability.**

**In Java, ‘this’ is a reference variable that refers to the current object, or can be said “this” in Java is a keyword that refers to the current object instance. It can be used to call current class methods and fields, to pass an instance of the current class as a parameter, and to differentiate between the local and instance variables. Using “this” reference can improve**

**code readability and reduce naming conflicts.**

**public class Person {**

**// Fields Declared**

**String name;**

**int age;**

**// Constructor**

**Person(String name, int age)**

**{**

**this.name = name;**

**this.age = age;**

**}**

**// Getter for name**

**public String get\_name() { return name; }**

**// Setter for name**

**public void change\_name(String name)**

**{**

**this.name = name;**

**}**

**// Method to Print the Details of**

**// the person**

**public void printDetails()**

**{**

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

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

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

**}**

**// main function**

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

**{**

**// Objects Declared**

**Person first = new Person("ABC", 18);**

**Person second = new Person("XYZ", 22);**

**first.printDetails();**

**second.printDetails();**

**first.change\_name("PQR");**

**System.out.println("Name has been changed to: "**

**+ first.get\_name());**

**}**

**}**

**2(a) - What is the use of String class in java ? Explain any of its three methods.**

**Ans - Strings, which are widely used in Java programming, are a sequence of characters. In the Java programming language, strings are objects.**

**The Java platform provides the String class to create and manipulate strings.**

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

**{**

**String str = new String("example");**

**// creating Java string by new keyword**

**// this statement create two object i.e**

**// first the object is created in heap**

**// memory area and second the object is**

**// created in String constant pool.**

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

**}**

**}**

**Q2(b) - Why are exception in java ? How can we handle these ?**

**Ans - class Main {**

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

**try {**

**// code that generate exception**

**int divideByZero = 5 / 0;**

**System.out.println("Rest of code in try block");**

**}**

**catch (ArithmeticException e) {**

**System.out.println("ArithmeticException => " + e.getMessage());**

**}**

**}**

**}**

**Q3(a)- What do you understand by constructor overloading ? Explain with example how it is different from method overiding?**

**Ans - class Box**

**{**

**double width, height,depth;**

**// constructor used when all dimensions**

**// specified**

**Box(double w, double h, double d)**

**{**

**width = w;**

**height = h;**

**depth = d;**

**}**

**// compute and return volume**

**double volume()**

**{**

**return width \* height \* depth;**

**}**

**}**

**import java.io.\*;**

**class MethodOverloadingEx {**

**static int add(int a, int b) { return a + b; }**

**static int add(int a, int b, int c)**

**{**

**return a + b + c;**

**}**

**Q3(b)- How can we final keyword in java ? what are arrays of objects ?**

**Ans- The final keyword can be applied with the variables, a final variable that have no value it is called blank final variable or uninitialized final variable. It can be initialized in the constructor only. The blank final variable can be static also which will be initialized in the static block only. We will have detailed learning of these. Let's first learn the basics of final keyword.**

**class Bike9{**

**final int speedlimit=90;//final variable**

**void run(){**

**speedlimit=400;**

**}**

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

**Bike9 obj=new Bike9();**

**obj.run();**

**Java is an object-oriented programming language. Most of the work done with the help of objects. We know that an array is a collection of the same data type that dynamically creates objects and can have elements of primitive types. Java allows us to store objects in an array. In Java, the class is also a user-defined data type. An array that conations class type elements are known as an array of objects. It stores the reference variable of the object.**

**class Student {**

**public int id;**

**public String name;**

**// Student class constructor**

**Student(int id, String name)**

**{**

**this.id = id;**

**this.name = name;**

**}**

**// display() method to display**

**// the student data**

**public void display()**

**{**

**System.out.println("Student id is: " + id + " "**

**+ "and Student name is: "**

**+ name);**

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

**}**

**}**

**Q.4(a)- What are 2d arrays in java ? write a program to print sum of all positive integer arrays values?**

**Ans- class MultidimensionalArray {**

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

**// create a 2d array**

**int[][] a = {**

**{1, 2, 3},**

**{4, 5, 6, 9},**

**{7},**

**};**

**// calculate the length of each row**

**System.out.println("Length of row 1: " + a[0].length);**

**System.out.println("Length of row 2: " + a[1].length);**

**System.out.println("Length of row 3: " + a[2].length);**

**}**

**}**

**public class SumOfArray {**

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

**//Initialize array**

**int [] arr = new int [] {1, 2, 3, 4, 5};**

**int sum = 0;**

**//Loop through the array to calculate sum of elements**

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

**sum = sum + arr[i];**

**}**

**System.out.println("Sum of all the elements of an array: " + sum);**

**}**

**}**

**Q4(b)- Write down the difference between instance variable and class varible and class variable and class variable with suitable example. Write a program to show working of static and non static blocks in java.**

**Ans- In Java, instance variables are variables declared within a class but outside of any method, constructor, or block. They are also known as member variables or fields. Instance variables hold values that are unique to each object created from the class. Here's an example of how you can declare and use instance variables in Java.**

**public class MyClass {**

**// Instance variables**

**private int number; // private variable can only be accessed within this class**

**public String text; // public variable can be accessed from any class**

**protected double value; // protected variable can be accessed within the package and subclasses**

**// Constructor to initialize the instance variables**

**public MyClass(int number, String text, double value) {**

**this.number = number;**

**this.text = text;**

**this.value = value;**

**}**

**// Getter and setter methods for accessing private instance variable**

**public int getNumber() {**

**return number;**

**}**

**public void setNumber(int number) {**

**this.number = number;**

**}**

**}**

**public class MyClass {**

**// Class variable**

**public static int classVariable = 0;**

**// Constructor**

**public MyClass() {**

**// Increment the class variable in the constructor**

**classVariable++;**

**}**

**// Method to access and modify the class variable**

**public static void printClassVariable() {**

**System.out.println("Class variable value: " + classVariable);**

**}**

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

**// Creating objects of MyClass**

**MyClass obj1 = new MyClass();**

**MyClass obj2 = new MyClass();**

**MyClass obj3 = new MyClass();**

**// Calling the method to print the class variable**

**MyClass.printClassVariable(); // Output: Class variable value: 3**

**}**

**}**