**Java: (day one)**

1)Steps to install java

Step1: Download the JDK, for that go to the [www.java.sun.com](http://www.java.sun.com)

Step2: Install JDK and extract the file .exe

Step3: Open CMD type JAVAC, not recognized

Step4: Go to Computer->Local Disk©->Program file->java->JDK->Bin->Applet viewer (In that right click)->Properties->Location (Copy the location)

Step5: Go ahead computer click on properties and advanced system settings->click on environment Variable->User variable->Click on new and variable name(Path)->Variable value is (copy paste location on step4)

Step6: Click on okay.

Step7: Now go to CMD and type JAVAC, compiler is ready.

2)Steps to install eclipse

Step1: Download the eclipse from oracle web site

Step2: Extract the folder

Step3: Click on install

Step4: Save the app in the applications for steps to create workspace

Step5: File->Search Work Space->Create work space by its name

3)Steps to create project and .java file/class

Step1: File->New->Project->Project Name->Finish

Step2: Create .java file/class

Step3: File->New->Class file->Java

Class file should be always being with upper case letter

4)How to create packages and what is best way to given name

File->New ->package

Package name should be always being with lower case letter

5) What is main method will do

Main is the single point of entry to most java programs.

1. Each statement in the main executes in order until the end of main is reached -- this is when program terminates;

6)What is data type and different data types

Data type specifies the size and type of values that can be stored in an identifier.

Data types in java are classified into 2 types.

1.Premitive: Which include integer, Character, Boolean and Floating point.

2.Non-Premitive: Which includes class, interfaces, and arrays.

7)What is variable?

A java variable is a piece of memory that can contain a data value.

There are different types of variables in java.

1.Instance Variables

2.Class Variables

3.Local Variables

8)Creating Method with void

Public static void main(String[] args){

9)Create variable, we can create variables inside method?

Variable can be created by associating a data type along with it we can’t create a static variable inside a method.

10)Creating method with return data type, we can return int/string/double/float/data etc

Yes. we can return a value from all specific method

Note: Value that we specify after return keyword should be specified in method signature.

Public class Length{

int length,breadth;

void setLength( int len){

length= len;

}

Int getLength()

{

Return length;

}}

11)Method that will return hard coded value?

Hard coding is not only the part of java. It comes when we put the original variables and data values in place of hard coding.

12)Create default/parametrized constructors?

When we don’t declare a constructor then a default constructor is created when an object is declared passed in it then it’s called as a parametrized constructor.

13)Creating method with return data type and parameter?

Public class Length{

int length,breadth;

void setLength( int len){

length= len;

}

Int getLength()

{

Return length;

}}

Parameter

Parameters should be passed in sequence and they must be accepted by method in the same sequence.

Void setParameters(String str,int len)

{

---

--

}

R1.setParameters(12,”Pritesh”);

14) creating static property?

The Static property in java is used for memory management mainly. We can apply java static keyword will methods, blocks and nested class. The static keyword belongs to the class than instance of the class.

15) creating static method

Java static method program: static methods in Java can be called without creating an object of class.

**class** Languages {

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

display();

}

**static** **void** display() {

System.out.println("Java is my favorite programming language.");

}

}

17) **Create static block?**

public class MyStaticBlock{

Public static void main(String a[]){

MyStaticBlock msb = new MyStaticBlock();

Msb.testList();

}

}

18) creating object

In this example, we have created a Student class that have two data members id and name. We are creating the object of the Student class by new keyword and printing the objects value.

**package** saibersys;

**public** **class** Student {

**int** id;

String name;

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

Student s1=**new** Student();

System.***out***.println(s1.id);

System.***out***.println(s1.name);

}

}

19) calling method with void

**package** saibersys;

**public** **class** ExampleMinNumber {

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

**int** a = 11;

**int** b = 6;

**int** c = *minFunction*(a, b);

System.***out***.println("Minimum Value = " + c);

}

20) Calling method with no return and parameter?

**package** saibersys;

**public** **class** Add{

**int** a=1;

int b=2;

c=a+b;

Add a+b();

}

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

System.***out***.println("Addition :”,c);

}

Output:- Addition: 3

21)Calling method with return and no parameter?

**package** saibersys;

**public** **class** Add{

**int** a=1;

int b=2;

c=a+b;

Add a+b();

}

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

return(c);

}

22)Calling method with return and parameter?

double total(double aNumber);

{

double a\_Value = aNumber+40;

return a\_value;}

doublel(double aNumber)

{

double a\_Value = aNumber+60;

return a\_Value;

}

23)Calling method with return and storing the return data?

**public** **class** PerfectNumber {

public boolean PerfectNumber(int number){

int temp=2;

for(int i=1;i<=number/2;i++){

if(number%1==0){

temp+=i;

}

}

if(temp==number){

System.out.println("It is a perfect number");

return true;

}

else{

System.out.println("It is not a perfect number");

return false;

}

//return value;

}

24) using static property: it will maintain

Public static int DayNumber;

25) create classes under multiple packages

Under single java project we can create multiple classes and multiple packages.

26)Calling classes under different packages

If access modifier for a class is Public, then it can be called by creating an object to it.

27)Write code to handle exceptions with try/catch/finally?

**package** saibersys;

**import** java.io.File;

**import** java.io.FileReader;

**import** java.io.IOException;

**public** **class** ReadData\_Demo {

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

FileReader fr=**null**;

**try**{

File file=**new** File("file.txt");

fr = **new** FileReader(file); **char** [] a = **new** **char**[50];

fr.read(a); // reads the content to the array

**for**(**char** c : a)

System.***out***.print(c); //prints the characters one by one

}**catch**(IOException e){

e.printStackTrace();

}

**finally**{

**try**{

fr.close();

}**catch**(IOException ex){

ex.printStackTrace();

}

}

}}

**28) what is checked exception/unchecked exception**

Anything that is a subclass of Exception except for Runtime Exception and its subclasses is a checked exception

**29)What is final keyword, create final class, final method, final property**

Final Key word is mainly used to restrict the java variable

**Public class** Car{

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

**void** run(){

 speedlimit=400;

}

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

 Car9 obj=**new**  Car9();

obj.run();

}  }

|  |
| --- |
| **30)write code for creating abstract class?**  public abstract class Employee {  private String name;  private String address;  private int number;  public Employee(String name, String address, int number) {  System.out.println("Constructing an Employee");  this.name = name;  this.address = address;  this.number = number;  }public double computePay() {  System.out.println("Inside Employee computePay");  return 0.0;  } public void mailCheck() {  System.out.println("Mailing a check to " + this.name + " " + this.address);  }public String toString() {  return name + " " + address + " " + number;  }public String getName() {  return name;  }    public String getAddress() {  return address;  }public void setAddress(String newAddress) {  address = newAddress;  } public int getNumber() {  return number;  }  }/Implementing the above Abstract demo |
| public class AbstractDemo {  public static void main(String [] args) {  Employee e = new Employee("George W.", "Houston, TX", 43);  System.out.println("\n Call mailCheck using Employee reference--");  e.mailCheck();  }  }  31)Implement method overloading?  class MyClass {  int height;  MyClass() {  System.out.println("bricks");  height = 0;  }  MyClass(int i) {  System.out.println("Building new House that is " + i + " feet tall");  height = i;  }  void info() {  System.out.println("House is " + height + " feet tall");  }  void info(String s) {  System.out.println(s + ": House is " + height + " feet tall");}  }  public class MainClass {  public static void main(String[] args) {  MyClass t = new MyClass(0);  t.info();  t.info("overloaded method");  //Overloaded constructor:  new MyClass();}}  32)Implement method overriding  class Man{  public void eat()  {  System.out.println("Human is eating");  }  }  class Boy extends Man{  public void eat(){  System.out.println("Boy is eating");  }  public static void main( String args[]) {  Boy obj = new Boy();  obj.eat();  }  }  33)Implementing polymorphism  Through method overloading and overriding.  34) Implementing interface?  Implementing interface with the Abstract class  35)How to do inheritance in java (using extend keyword)  class MyBaseClass{  protected void disp()  {  System.out.println("Parent class method");  }  }  class MyChildClass extends MyBaseClass{ //thhis is the way when we use inheritance //  public void disp(){  System.out.println("Child class method");  }  public static void main( String args[]) {  MyChildClass obj = new MyChildClass();  obj.disp();  }  }  36) Write code to add items to integer, string array  public static void main(String[] args) {  int[] series = new int[0];  int x = 5;  series = addInt(series, x); System.out.print("New series: ");  for (int i = 0; i < series.length; i++){  if (i == series.length - 1){  System.out.println(series[i]);  }  else{System.out.print(series[i] + ", ");}}}  37)write code to retrieve items from integer, string array  public int getArrayIndex(int[] arr,int value) {  int k=0;  for(int i=0;i<arr.length;i++){  if(arr[i]==value){  k=i;  break;}}  return k;  }  38)write code to add items to ArrayList collection  import java.util.\* ;  public class ArrayList  {  public static void main ( String[] args)  {  ArrayList<String> names = new ArrayList<String>();  System.out.println("initial size: " + names.size() );  names.add("prasu");  names.add("vasu");  names.add("bhavana");  System.out.println("new size: " + names.size() );  for ( int j=0; j<names.size(); j++ )  System.out.println("element " + j + ": " + names.get(j) );}}  40)write code to retrieve items from arraylist (using for each loop\_  **private** **void** **Customer Info form** (**object** sender, EventArgs e)  {  ArrayList arrayList = **new** ArrayList();  arrayList.Add("Customer1");  arrayList.Add("Customer2");  arrayList.Add("Customer3");  arrayList.Add("Customer4");  arrayList.Add("Customer5");  **string** str = **string**.Empty;  **foreach** (**string** strName **in** arrayList)  {  str += strName + "\n";  }    41)write code to add items HashMap  hm.put("A", new Person("p1"));  hm.put("B", new Person("p2"));  hm.put("C", new Person("p3"));  hm.put("D", new Person("p4"));  hm.put("E", new Person("p5"));  Set<Map.Entry<String, Person>> set = hm.entrySet();  for (Map.Entry<String, Person> me : set) {  System.out.println("Key :"+me.getKey() +" Name :  "+ me.getValue().getName()+"Age :"+me.getValue().getAge());  }  42) **write code to retrieve items HashMap**  public class HashMapDemo {  public static void main(String args[]) {  // create hash map  HashMap newmap = new HashMap();  // populate hash map  newmap.put(1, "tutorials");  newmap.put(2, "point");  newmap.put(3, "is best");  // get value of key 3  String val=(String)newmap.get(3);  // check the value  System.out.println("Value for key 3 is: " + val);  }  }  **43)write code to connect to JDBC to get rows from employee table**  44) **Write method to return list of rows code to loop throughs**  **private** **void** **Customer Info form** (**object** sender, EventArgs e)  {  ArrayList arrayList = **new** ArrayList();  arrayList.Add("Customer1");  arrayList.Add("Customer2");  arrayList.Add("Customer3");  arrayList.Add("Customer4");  arrayList.Add("Customer5");  **string** str = **string**.Empty;  **foreach** (**string** strName **in** arrayList)  {  str += strName + "\n";  }  45)Create Employee class  **package** saibersys;  **public** **class** Employee {  String lastName = **null**;  String firstName = **null**;  **double** ID;  **public** Employee(String lastName, String firstName, **double** ID){  **this**.lastName = lastName;  **this**.firstName = firstName;  **this**.ID = ID;  }  **public** String empStat(){  **return** "Last Name: " + lastName + "First Name: " + firstName + "ID" + ID;  }}  46) **create method that return list of employee collection**  public class Employee {  int empid;  String name;  int age;  public Employee(int empid,String name,int age)  {  this.empid=empid;  this.name=name;  this.age=age; }  public int getEmpid() {  return empid;  }  public void setEmpid(int empid) {  this.empid = empid;  }  public String getname() {  return name;  }  public void setname(String name) {  this.name = name;  }  public int getAge() {  return age;  }  public void setAge(int age) {  this.age = age;  }}  comparator class:  public class Employee\_comparator implements Comparator<Employee> {  @Override  public int compare(Employee object1, Employee object2) {  return object1.getname().compareTo(object2.getname());  }  }  47)Difference between string, string buffer, string builder with example  String: String class is immutable.  String is slow and consumes more memory when you contact too many strings because every time it creates  new instance  String class overrides the equals () method of object class.  String Buffer: String Buffer class is mutable.  String Buffer is fast and consumes less memory  String Builder: Java StringBuilder class is used to create mutable (modifiable) string.  48)how to update the data into XML file and read data from XML file  We can update using web.xml or pmo.xml  <?xml version="1.0" encoding="UTF-8" standalone="no"?>  <data>  <username>admin</username>  <password>12345</password>  <interval>1</interval>  <timeout>90</timeout>  <startdate>01/01/2013</startdate>  <enddate>06/01/2013</enddate>  <ttime>1110</ttime>  </data>   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

|  |
| --- |
|  |
|  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |