|  |  |  |
| --- | --- | --- |
| **Name:Pooanamchand sahu** | **Emp.ID:** | **Subject:** |
| **Company Name:** | **Max Marks Obtained:** | **Actual Marks:** |
| **Trainer Name:** | **Duration:** | **Email-Id:** |

**Note: *Overwriting is disallowed; any overwriting will be subject to deduction of Marks.***

**1.**

**public class Student10{**

**int id;**

**String name;**

**Student10(int id,String name){**

**id = id;**

**name = name;**

**}**

**void display(){System.*out*.println(id+" "+name);}**

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

**Student10 s1 = new Student10(111,"Karan");**

**Student10 s2 = new Student10(321,"Aryan");**

**s1.display();**

**s2.display(); } }**

**A) 111 karan,321 Aryan B)321 Aryan,111 karan C)compile time error D)0 null,0 null**

**Ans: {D}-1**

**2.**

**class Test{ class Test{**

**p s v main(String args[]) p s v main(String []args)**

**{ {**

**int x =10; boolean x =false;**

**if(x=20) if(x=false)**

**sopln("hello"); sopln("hello");**

**else else**

**sopln("hi");} sopln("hi");}**

**A) hello, hi B) hi, compile time error C) compile time error , hello D) hi,hi**

**Ans : {D}**

**3. We can override a private,static,final method in Java?**

**A) Yes B) No C) Only static D Only private**

**Ans: {B}-1**

**4. We can overload a static method in Java?**

**A) Yes B) No**

**Ans : {A}-1**

**5.**

**int i=0;**

**for(sopln("hello");i<3;sopln("hi"))**

**{**

**i++;**

**}**

1. **Valid B) Invalid**

**Ans : {A}-1**

**6. How can such a restriction be enforced ?**

A method within a class is only accessible by classes that are defined within the same package as the class of the method.

|  |  |
| --- | --- |
| **A.** Declare the method with the keyword public | **B.** Declare the method with keyword protected |
| **C.** Declare the method with keyword private | **D.** Without any accessibility specifiers. |

**Ans:-{D}-1**

**7. A special method that is used to initialize a class object ?**

|  |  |
| --- | --- |
| **A.** abstract method | **B.** static method |
| **C.** Constructor | **D.** overloaded method. |

**Ans: {C}-1**

**8. class A**

**{**

**A()**

**{**

**sopln("A constructor");**

**} }**

**class B extends A**

**{**

**B()**

**{**

**//Super();**

**sopln("B constructor");**

**}**

**p s v main(String args[])**

**{**

**B b=new B();**

**}}**

1. **Compile time error B) Exception C) B constructor D) None**

**Ans: {D}-1**

**9. int x=10; int x=10;**

**switch(x+1) int y=20;**

**{ switch(x) {**

**case 10: case 10:**

**sopln(10); sopln(10);**

**break; break;**

**case 10+20+30: case y:**

**sopln(60);break; sopln(20);**

**case 11: break;**

**sopln(11); }**

**break;**

**}**

**A). 11, 20 B) compile time error , compile time error C)11,compile time error**

**D) compile time error,20**

**Ans: {C}-1**

**10. Can we prevent overriding a method without using the final modifier?  
A) if yes how ? B) if no why ?**

**Ans:**

**Yes we can prevent overriding a method by declaring it as private.-1**

**11. Can we override a private method in Java?  
a) yes b)no**

**Ans: {B}-1**

**12.Abstract classes consist of Constructors?**

**a) yes b) no**

**Ans: {B}**

**13. How many methods present in Object class?**

**a) zero b) 12 c) 11 d) None**

**Ans: {c}-1**

**14. Can we change the argument list of an overriding method?**

1. **yes b)no**

**Ans: {B}-1**

##### **15. Garbage collection in Java is**

**A) Unused package in a program automatically gets deleted.**

**B) Memory occupied by objects with no reference is automatically reclaimed for deletion.**

**C) Java deletes all unused java files on the system.**

1. **The JVM cleans output of Java program.**

**Ans: {B}-1**

**16. Can we make a class abstract without an abstract method?  
A) Yes B) No**

**Ans: {A}-1**

**17. Which of these process occur automatically by java run time system?**

**A) Serialization B)Garbage Collector C) FileFiltering D)All the above**

**Ans: {B}-1**

**18)Which of the following regarding Abstract classes are true?**

**a)An abstract class can be extended.**

**b)A sub class of non abstract sub class can be abstract.**

**c)A sub class can override a concrete method in super class to declare it as abstract.**

**d)An abstract class can be used as a data type.**

**e)All the above.**

**Ans: {e}-1**

**19)What happens if don’t assign any values to variables in interfaces.**

**A) Run successfully B) Compile time error C) Assigned with default values D) None of above**

**Ans: {B}-1**

**20. When the methods can be override in below cases?**

**1) Should have same return type and same method name.**

**2) Return type can be different in case if their return type is non primitive.**

**3) Can have a same method name with different return types.**

**4) Can have same method name ,numbers of parameters need not be same.**

1. **1&3 B) 2&3 C) 1&4 D) 1&2**

**Ans: {B}**

**21. The default value of a static integer variable of a class in Java is,**

**A)  0                       B) 1                       C) Garbage value   D) Null    E) -1**

**Ans: {A}-1**

**22. What will be printed as the output of the following program?**

**public class Test**

**{**

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

**{ int i = 0;**

**i = i++ + i;**

**System.out.println("I = " +i);**

**} }**

**Ans: 1-1**

**23. Multiple inheritance means,**

**(a)   one class inheriting from more super classes**

**(b)   more classes inheriting from one super class**

**(c)   more classes inheriting from more super classes**

**(d)   None of the above**

**(e)   (a) and (b) above.**

**Ans: {a}-1**

**24. Which statement is not true in java language?**

**(a)   A public member of a class can be accessed in all the packages.**

**(b)   A private member of a class cannot be accessed by the methods of the same class.**

**(c)   A private member of a class cannot be accessed from its derived class.**

**(d)   A protected member of a class can be accessed from its derived class.**

**(e)   None of the above.**

**Ans: {b}-1**

**25.Difference between ‘= =’, String.equals(), compareTo() ?**

**Ans : ‘==’ checks if two reference point to the same object,**

**String.equals() checks if two string have the same value or not**

**compareTo() it calculates the difference between two strings and returns 0 if they are same else it returns the difference.**

**26.**

**String s=”welcome”;**

**Sopln(s.codePointBefore(3));**

**Sopln(s.subString(1,3));**

**A)compile time error B) 108,elc C) 108,el D)108,welcome**

**Ans: {C}-1**

**27.**

**class test**

**{**

**public void m1()**

**{ sopln("no arg"); }**

**public void m1(int i)**

**{ sopln("int org"); }**

**p s v main(string[] args)**

**{**

**test t=new test();**

**t.m1();**

**t.m1(10);**

**t.m1(10.5);}} OutPut:-**

**Ans :**

**Compile time error -1**

**28.**

**String s=”welcome”;**

**String s1=new String(”welcome”);**

**StringBuffer s2=new StringBuffer(“welcome”);**

**StringBuffer s3=new StringBuffer(“welcome”);**

**Sopln(s.equeals(s1));**

**Sopln(s2.equeals(s3));**

**Sopln(s==s1));**

1. **True,True,True B) False,False,False C) False,True,True D) True,False,False E)True,True,False**

**Ans: {D} -1**

**29) Write any one difference between String, StringBuffer, StringBuilder ?**

**Ans : String is immutable and not threadsafe whereas StringBuilder is mutable and not threadsafe and StringBuffer is mutable as well as thread safe.**

**30) Explain the functionalities of three different loaders in JVM ?**

**The 3 loaders are :**

1. **Bootstrap class loader :It loads the rt.jar file which contains all the Java API classes into the JVM initial class.**
2. **Extension class loader : It loads the extension classes that are present inside the ‘ext’ directory into the JVM.**
3. **Application class loader : It loads all the user defined and other classes that are present in the class path and environment path variables.**