1. Which of following statements about Java language is true? (A)

A. Both procedural and OOP are supported in Java.

* 1. Java supports only procedural approach towards programming.
  2. Java supports only OOP approach.
  3. None of the above.

1. If an attribute has no access specifier (default), which methods have access to it? A. Those defined in the same class. (D)
   1. Static methods in the same class.
   2. Classes in the same package.
   3. A and C
2. Analyze the code snippet below and determine the output: (D) **package** com.core.ct;  **class** One{

**private** One(){

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

}

**public** **void** display(){ System.*out*.println("Display");

}

}

**public** **class** Two **extends** One{

**public** Two(){

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

}

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

|  |  |  |
| --- | --- | --- |
|  |  | **new** Two(); |
|  |  | One obj=**new** Two(); |
|  |  | obj.display(); |
|  |  |  |
| } | } |  |
|  | A. One | Two Display |
|  | B. Two | One Display |
| Ii0 | C. One | Display Two |

D. Compilation Error

4. select the missing piece of code to ensure successful compilation: (B)

**package** com.core.ct;  **class** One{

**public** One(String name){

}

}

**public** **class** Two **extends** One{

**public** Two(){ // Insert Code Here

}

}

A. One(“name”) B. super.One(“name”) C.super(“name”);

D. this(“name”);

5. Analyze the code snippet below and determine the output: (C)

**package** com.core.ct;

**class** One {

**private** **final** **void** showMessage() { System.*out*.println("One show method");

}

}

**public** **class** Two **extends** One { **public** **final** **void** showMessage() { System.*out*.println("Two show method");

}

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

Two obj = **new** Two();

obj.showMessage();

}

}

A. One show method B. Two show method C. Compilation Error D. Runtime Error

6. determine the output: (C)

**package** com.core.ct;

**class** One {

**public** **void** showMessage1() { System.*out*.println("Method-1");

}

}

**public** **class** Two **extends** One { **public** **void** showMessage2() { System.*out*.println("Method-2");

}

**public** **void** showMessage1(){ System.*out*.println("Method-3");

}

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

One ob1 = **new** Two(); Two ob2 = (Two)ob1; ob2.showMessage1(); ob2.showMessage2();

}

}

1. Method - 1 Method – 2

1. Method – 2

Method – 1

1. Method – 3

Method – 2

1. Compilation Error

**7.** Analyze the code snippet below and determine the output: (A)

**package** com.core.ct;  **class** One {

**int** a=100;

}

**public** **class** Two **extends** One {

**int** a=999;

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

One ob=**new** Two();

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

}

}

A. 100 B. 999 C. Compilation Error D. Runtime Exception

8. determine the output:

**package** com.core.ct; (C)

**final** **class** One {

**public** **void** showMessage1() { System.*out*.println("Method-1");

}

}

**public** **class** Two **extends** One { **public** **void** showMessage1(){ System.*out*.println("Method-2");

}

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

One ob1 = **new** Two();

ob1.showMessage1();

}

}

A. Method -1 B. Method – 2 C. Compilation Error D. Runtime Exception

9. Analyze the code snippet below and determine the output: (B)

**package** com.core.ct;  **class** One { }

**class** Two **extends** One { }

**public** **class** TestDemo {

**boolean** method1(One one) {

**return** **true**;

}

**boolean** method1(Two two) { **return** **false**;

}

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

One ob1 = **new** One();

Two ob2 = **new** Two();

TestDemo obj = **new** TestDemo();

System.*out*.println(obj.method1(ob1));

System.*out*.println(obj.method1(ob2));

}

}

1. false true
2. true false
3. The program generates compilation error
4. The program generates runtime exception

1. Consider the following statements: (C)

* 1. The keyword ‘super’ is used by subclass to invoke an overridden method in the

superclass

* 1. Class can have only a constructor
  2. For every class , super class is Class
  3. Final methods can’t be overridden in the subclass

Which of these are true?

* 1. 2
  2. 1 & 2
  3. 1 & 4
  4. 3

1. Analyze the code snippet below and fill in the blanks to ensure successful compilation.(D)

**package** com.core.ct;

**class** One {

**public** **void** showMessage(Object o){ System.*out*.println("Method-1");

}

}

**class** Two **\_\_\_\_\_\_\_\_\_\_\_** One {

\_\_\_\_\_\_\_\_\_\_ void showMessage(\_\_\_\_\_\_\_\_\_\_\_\_ n){

System.*out*.println("Method -2");

}

}

A. implements protected String B. implements public String

1. extends protected String
2. extends public String

1. Analyze the code snippet below and select the missing piece of code to ensure successful compilation: (D)

**package** com.core.ct;

**class** One {

**public** **int** showMessage(**int** o){

**return** 0;

}

}

**class** Two **extends** One {

// Insert Code here

}

* 1. public long showMessage(int i){

return 1;

}

* 1. protected int showMessage(int i){ return 1;

}

* 1. private int showMessage(int i){ return 1;

}

* 1. public int showMessage(int i){ return 1;

}

1. A class can have many methods with the same name, as long as the number of parameters or type of parameters are different. This OOP concept is known as (D)
   * 1. Method Invocating
     2. Method Overriding
     3. Method Labeling
     4. Method Overloading

1. Instantiation is completed when (A)
   * 1. Memory is allocated for a specific object of a class.
     2. java Command is executed
     3. A program is ready for execution.
     4. A program compiles correctly.

1. Inheritance is the process of (B)
   * 1. Using classes in the established standard Java Language library.
     2. Using features from an existing class.
     3. Combining data and the methods, which process the data, inside the same module.
     4. Splitting a program into multiple related files for each class in the program.
2. What is the output of the following code snippet? (A)

class One{

One(){

System.out.print("bike ");

}

One(String s){ this();

System.out.print("car ");

}

}

class Two extends One{

Two(){

System.out.print("jeep ");

}

Two(String s){

this();

System.out.print(s+" ");

}

}

public class Demo { public static void main(String[] args) { Two obj=new Two("are under repair");

}

}

1. bike car jeep are under repair
2. jeep bike car are under repair
3. bike jeep car are under repair
4. bike jeep are under repair
5. jeep bike are under repair

17. What is the output of the following code snippet? (C)

class Person{

String name="VP"; public String speaks(){

return "speaks Telugu";

}

}

class Employee extends Person{

String name="ALN"; public String speaks(){

return "speaks Hindi";

}

}

public class Demo { public static void main(String[] args) {

new Demo().getInfo();

}

public void getInfo(){

Person person=new Employee();

System.out.println(person.name+" "+person.speaks());

}

}

1. ALN speaks Hindi
2. VP speaks Telugu
3. ALN speaks Telugu
4. VP speaks Hindi
5. None of the above

18. What is the output of the following code snippet? (C)

class One{

static String message=""; protected One(){ message+="I ";

}

}

class Two extends One{ private Two(){

message+="the";

}

}

public class Demo extends One { private Demo(){

this("am ");

message+="King ";

}

private Demo(String s){

message+=s;

}

public static void main(String[] args) {

Demo d=new Demo();

System.out.println(d.message);

}

}

1. I am the King
2. I King
3. I am King
4. Am I king
5. King I am

19. What is the output of the following code snippet?class One{

public String getMessage(){

return "I am from one";

}

}

class Two extends One{

public String getMessage(){

return "I am from two";

}

public String getGreetings(){

return "Hi how are you?";

}

}

public class Demo extends One { public static void main(String[] args) {

One obj1=new Two();

Two obj2=(Two)obj1;

System.out.println(obj2.getGreetings());

One obj3=new Two();

Two obj4=(Two)(One)(Two)obj3;

System.out.println(obj4.getMessage()); (B)

}

}

A. Hi how are you? I am from One B. Hi how are you? I am from Two

1. I am from One

Hi how are you?

1. Hi how are you?
2. None of the above

20. What is the output of the following code snippet? (C)

class One{

int a=100;

public static String printMessage(){

return "Hello";

}

}

class Two extends One{

int a=200;

public static String printMessage(){

return "Hi";

}

}

public class Demo extends One { public static void main(String[] args) {

One obj=new Two();

System.out.println(obj.a+" "+obj.printMessage());

}

}

1. 200 Hello
2. 100 Hi
3. 100 Hello
4. 200 Hi
5. None of the above