Java Fundamentals Assessment Test -1

1. What is the output for the below code ?

1. public class A {

2. int add(int i, int j){

3. return i+j;

4. }

5.}

6.public class B extends A{

7. public static void main(String argv[]){

8. short s = 9;

9. System.out.println(add(s,6));

10. }

11.}

Options are

A.Compile fail due to error on line no 2

B.Compile fail due to error on line no 9

C.Compile fail due to error on line no 8

D.15

Answer :B is the correct answer

Explanation:Cannot make a static reference to the non-static method add(int, int) from the type A. The short s is autoboxed correctly, but the add() method cannot be invoked from a static method because add() method is not static

2)What is the output for the below code ?

public class A {

int k;

boolean istrue;

static int p;

public void printValue() {

System.out.print(k);

System.out.print(istrue);

System.out.print(p);

}

}

public class Test{

public static void main(String argv[]){

A a = new A();

a.printValue();

}

}

Options are

A.0 false 0

B.0 true 0

C.0 0 0

D.Compile error - static variable must be initialized before use.

Answer : A is the correct answer.

Explanation:Global and static variable need not be initialized before use. Default value of global and static int variable is zero. Default value of boolean variable is false. Remember local variable must be initialized before use.

3)What is the output for the below code ?

public class Test{

int \_$;

int $7;

int do;

public static void main(String argv[]){

Test test = new Test();

test.$7=7;

test.do=9;

System.out.println(test.$7);

System.out.println(test.do);

System.out.println(test.\_$);

}

}

Options are

A.7 9 0

B.7 0 0

C.Compile error - $7 is not valid identifier.

D.Compile error - do is not valid identifier.

Answer : D is the correct answer.

Explanation:$7 is valid identifier. Identifiers must start with a letter, a currency character ($), or underscore ( \_ ). Identifiers cannot start with a number. You can't use a Java keyword as an identifier. do is a Java keyword.

4)What is the output for the below code ?

package com;

class Animal {

public void printName(){

System.out.println("Animal");

}

}

package exam;

import com.Animal;

public class Cat extends Animal {

public void printName(){

System.out.println("Cat");

}

}

package exam;

import com.Animal;

public class Test {

public static void main(String[] args){

Animal a = new Cat();

a.printName();

}

}

Options are

A.Animal

B.Cat

C.Animal Cat

D.Compile Error

Answer :D is the correct answer.

Explanation:Cat class won't compile because its superclass, Animal, has default access and is in a different package. Only public superclass can be accessible for different package.

5)What is the output for the below code ?

public class A {

int i = 10;

public void printValue() {

System.out.println("Value-A");

};

}

public class B extends A{

int i = 12;

public void printValue() {

System.out.print("Value-B");

}

}

public class Test{

public static void main(String argv[]){

A a = new B();

a.printValue();

System.out.println(a.i);

}

}

Options are

A.Value-B 11

B.Value-B 10

C.Value-A 10

D.Value-A 11

Answer :B is the correct answer.

Explanation:If you create object of subclass with reference of superclass like ( A a = new B();) then subclass method and super class variable will be executed.

6)What is the output for the below code ?

public enum Test {

BREAKFAST(7, 30), LUNCH(12, 15), DINNER(19, 45);

private int hh;

private int mm;

Test(int hh, int mm) {

assert (hh >= 0 && hh <= 23) : "Illegal hour.";

assert (mm >= 0 && mm <= 59) : "Illegal mins.";

this.hh = hh;

this.mm = mm;

}

public int getHour() {

return hh;

}

public int getMins() {

return mm;

}

public static void main(String args[]){

Test t = new BREAKFAST;

System.out.println(t.getHour() +":"+t.getMins());

}

}

Options are

A.7:30

B.Compile Error - an enum cannot be instantiated using the new operator.

C.12:30

D.19:45

Answer :B is the correct answer.

Explanation:As an enum cannot be instantiated using the new operator, the constructors cannot be called explicitly. You have to do like

Test t = BREAKFAST;

7)What is the output for the below code ?

public class A {

static{System.out.println("static");}

}

Options are

A.A block static

B.static block A

C.static A

D.A

Answer :B is the correct answer.

Explanation:First execute static block, then statement block then constructor.

8)What is the output for the below code ?

1. public class Test {

2. public static void main(String[] args){

3. int i = 010;

4. int j = 07;

5. System.out.println(i);

6. System.out.println(j);

7. }

8. }

Options are

A.8 7

B.10 7

C.Compilation fails with an error at line 3

D.Compilation fails with an error at line 5

Answer : A is the correct answer.

Explanation:By placing a zero in front of the number is an integer in octal form. 010 is in octal form so its value is 8.

9)What is the output for the below code ?

1. public class Test {

2. public static void main(String[] args){

3. byte b = 6;

4. b+=8;

5. System.out.println(b);

6. b = b+7;

7. System.out.println(b);

8. }

9. }

Options are

A.14 21

B.14 13

C.Compilation fails with an error at line 6

D.Compilation fails with an error at line 4

Answer :C is the correct answer.

Explanation:int or smaller expressions always resulting in an int. So compiler complain about Type mismatch: cannot convert from int to byte for b = b+7; But b += 7; // No problem because +=, -=, \*=, and /= will all put in an implicit cast. b += 7 is same as b = (byte)b+7 so compiler not complain.

10)What is the output for the below code ?

public class Test {

public static void main(String[] args){

String value = "abc";

changeValue(value);

System.out.println(value);

}

public static void changeValue(String a){

a = "xyz";

}

}

Options are

A.abc

B.xyz

C.Compilation fails

D.Compilation clean but no output

Answer :A is the correct answer.

Explanation:Java pass reference as value. passing the object reference, and not the actual object itself. Simply assigning to the parameter used to pass the value into the method will do nothing, because the parameter is essentially a local variable.

11)What is the output for the below code ?

public class Test {

public static void printValue(int i, int j, int k){

System.out.println("int");

}

public static void printValue(byte...b){

System.out.println("long");

}

public static void main(String... args) {

byte b = 9;

printValue(b,b,b);

}

}

Options are

A.long

B.int

C.Compilation fails

D.Compilation clean but throws RuntimeException

Answer :B is the correct answer.

Explanation:Primitive widening uses the smallest method argument possible. (For Example if you pass short value to a method but method with short argument is not available then compiler choose method with int argument). But in this case compiler will prefer the older style before it chooses the newer style, to keep existing code more robust. var-args

method is looser than widen.

12)Fill in the gap:

public class Test {

public static void main(String[] args) {

String[] words = new String[] {"aaa", "bbb", "ccc", "aaa"};

Map<String, Integer> m = new TreeMap<String, Integer>();

for (String word : words) {

freq = m.get(word);

m.put(word, freq == null ? 1 : freq + 1);

}

System.out.println(m);

}

Use the following fragments zero or many times

String

Integer

Boolean

Float

13)You have a java file name Test.java inside src folder of javaproject

directory.You have also classes folder inside javaproject directory.

you have issued below command from command prompt.

cd javaproject

Which of the below command puts Test.class file inside classes folder ?

Options are

A.javac -d classes src/Test.java

B.javac Test.java

C.javac src/Test.java

D.javac classes src/Test.java

Answer :A is the correct answer.

Explanation:The -d option lets you tell the compiler in which directory to put the .class file it generates (d for destination)

14)You have two class files name Test.class and Test1.class inside

javaproject directory.

Test.java source code is :

public class Test{

public static void main (String[] args){

System.out.println("Hello Test");

}

}

Test1.java source code is :

public class Test1{

public static void main (String[] args){

System.out.println("Hello Test1");

}

}

you have issued below commands from command prompt.

cd javaproject

java Test Test1

What is the output ?

Options are

A.Hello Test

B.Hello Test Hello Test1

C.Hello Test1

D.Run fails - class not found

Answer :A is the correct answer.

Explanation:You must specify exactly one class file to execute. If more than one then first one will be executed.

15)You have a java file name Test.java .

Test.java needs access to a class contained in app.jar in "exam"

directory.Which of the following command set classpath to compile clean?

Options are

A.javac -classpath exam/app.jar Test.java

B.javac -classpath app.jar Test.java

C.javac -classpath exam Test.java

D.None of the above

Answer :A is the correct answer.

Explanation:javac -classpath exam/app.jar Test.java is the correct command to set exam/app.jar in classpath.

16)What will be the result of compiling the following code:

public class SuperClass {

public int doIt(String str, Integer... data)throws Exception{

String signature = "(String, Integer[])";

System.out.println(str + " " + signature);

return 1;

}

}

public class SubClass extends SuperClass{

public int doIt(String str, Integer... data)

{

String signature = "(String, Integer[])";

System.out.println("Overridden: " + str + " " +

signature);

return 0;

}

public static void main(String... args)

{

SuperClass sb = new SubClass();

sb.doIt("hello", 3);

}

}

Options are

A.Overridden: hello (String, Integer[])

B.hello (String, Integer[])

C.Compilation fails

D.None of the above

Answer :C is the correct answer.

Explanation: Unhandled exception type Exception.

17)What happens when the following code is compiled and run.

Select the one correct answer.

for(int i = 2; i < 4; i++)

for(int j = 2; j < 4; j++)

if(i < j)

assert i!=j : i;

Options are

A.The class compiles and runs, but does not print anything.

B.The number 2 gets printed with AssertionError

C.compile error

D.The number 3 gets printed with AssertionError

Answer :A is the correct answer.

Explanation:When if condition returns true, the assert statement also returns true. Hence AssertionError does not get generated. .

18)What happens when the following code is compiled and run.

Select the one correct answer.

for(int i = 2; i < 4; i++)

for(int j = 2; j < 4; j++)

assert i!=j : i;

Options are

A.The class compiles and runs, but does not print anything.

B.The number 2 gets printed with AssertionError

C.compile error

D.The number 3 gets printed with AssertionError

Answer :B is the correct answer.

Explanation:When i and j are both 2, assert condition is false, and AssertionError gets generated. .

19)

try{

File f = new File("a.txt");

}catch(Exception e){

}catch(IOException io){

}

Is this code create new file name a.txt ?

Options are

A.True

B.False

C.Compilation Error

D.None

Answer :C is the correct answer.

Explanation: IOException is unreachable to compiler because all exception is going to catch by Exception block.

20)

class A {

A(String s) {

}

A() {

}

}

1. class B extends A {

2. B() { }

3. B(String s) {

4. super(s);

5. }

6. void test() {

7. // insert code here

8. }

9. }

Which of the below code can be insert at line 7 to make clean

compilation ?

Options are

A.A a = new B();

B.A a = new B(5);

C.A a = new A(String s);

D.All of the above

Answer :A is the correct answer.

Explanation:A a = new B(); is correct because anonymous inner classes are no different from any other class when it comes to polymorphism

21)What is the output for the below code ?

interface A {

public void printValue();

}

1. public class Test{

2. public static void main (String[] args){

3. A a1 = new A() {

4. public void printValue(){

5. System.out.println("A");

6. }

7. };

8. a1.printValue();

9. }

10. }

Options are

A.Compilation fails due to an error on line 3

B.A

C.Compilation fails due to an error on line 8

D.null

Answer :B is the correct answer.

Explanation:The A a1 reference variable refers not to an instance of interface A, but to an instance of an anonymous (unnamed) class. So no compilation error.

22)

class A {

class A1 {

void printValue(){

System.out.println("A.A1");

}

}

}

1. public class Test{

2. public static void main (String[] args){

3. A a = new A();

4. // INSERT CODE

5. a1.printValue();

6. }

7. }

Which of the below code inserted at line 4, compile and produce the

output "A.A1"?

Options are

A.A.A1 a1 = new A.A1();

B.A.A1 a1 = a.new A1();

C.A a1 = new A.A1();

D.All of the above

Answer :B is the correct answer.

Explanation:correct inner class instantiation syntax is

A a = new A();

A.A1 a1 = a.new A1();

23)What is the output for the below code ?

public class A {

public void printValue(){

System.out.println("Value-A");

}

}

public class B extends A{

public void printNameB(){

System.out.println("Name-B");

}

}

public class C extends A{

public void printNameC(){

System.out.println("Name-C");

}

}

1. public class Test{

2. public static void main (String[] args) {

3. B b = new B();

4. C c = new C();

5. newPrint(b);

6. newPrint(c);

7. }

8. public static void newPrint(A a){

9. a.printValue();

10. }

11. }

Options are

A.Value-A Name-B

B.Value-A Value-A

C.Value-A Name-C

D.Name-B Name-C

Answer :B is the correct answer.

Explanation:Class B extended Class A therefore all methods of Class A will be available to class B except private methods. Class C extended Class A therefore all methods of Class A will be available to class C except private methods.

24)What is the output for the below code ?

public class A {

public void printName(){

System.out.println("Value-A");

}

}

public class B extends A{

public void printName(){

System.out.println("Name-B");

}

}

public class C extends A{

public void printName(){

System.out.println("Name-C");

}

}

1. public class Test{

2. public static void main (String[] args) {

3. B b = new B();

4. C c = new C();

5. b = c;

6. newPrint(b);

7. }

8. public static void newPrint(A a){

9. a.printName();

10. }

11. }

Options are

A.Name-B

B.Name-C

C.Compilation fails due to an error on lines 5

D.Compilation fails due to an error on lines 9

Answer : C is the correct answer.

Explanation:Reference variable can refer to any object of the same type as the declared reference OR can refer to any subtype of the declared type. Reference variable "b" is type of class B and reference variable "c" is a type of class C. So Compilation fails.

25)What is the output for the below code ?

public class C {

}

public class D extends C{

}

public class A {

public C getOBJ(){

System.out.println("class A - return C");

return new C();

}

}

public class B extends A{

public D getOBJ(){

System.out.println("class B - return D");

return new D();

}

}

public class Test {

public static void main(String... args) {

A a = new B();

a.getOBJ();

}

}

Options are

A.class A - return C

B.class B - return D

C.Compilation fails

D.Compilation succeed but no output

Answer :B is the correct answer.

Explanation:From J2SE 5.0 onwards. return type in the overriding method can be same or subtype of the declared return type of the overridden (superclass) method.

26)What is the output for the below code ?

public class A {

private void printName(){

System.out.println("Value-A");

}

}

public class B extends A{

public void printName(){

System.out.println("Name-B");

}

}

public class Test{

public static void main (String[] args) {

B b = new B();

b.printName();

}

}

Options are

A.Value-A

B.Name-B

C.Value-A Name-B

D.Compilation fails - private methods can't be override

Answer :B is the correct answer.

Explanation:You can not override private method , private method is not availabe in subclass . In this case printName() method a class A is not overriding by printName() method of class B. printName() method of class B different method. So you can call printName() method of class B.

27)What is the output for the below code ?

import java.io.FileNotFoundException;

public class A {

public void printName() throws FileNotFoundException {

System.out.println("Value-A");

}

}

public class B extends A{

public void printName() throws NullPointerException{

System.out.println("Name-B");

}

}

public class Test{

public static void main (String[] args) throws Exception{

A a = new B();

a.printName();

}

}

Options are

A.Value-A

B.Compilation fails-Exception NullPointerException is not compatible with throws

clause in A.printName()

C.Name-B

D.Compilation succeed but no output

Answer :C is the correct answer.

Explanation:The overriding method can throw any unchecked (runtime) exception, regardless of exception thrown by overridden method. NullPointerException is RuntimeException so compiler not complain.

28)What is the output for the below code ?

public class A {

public A(){

System.out.println("A");

}

public A(int i){

this();

System.out.println(i);

}

}

public class B extends A{

public B (){

System.out.println("B");

}

public B (int i){

this();

System.out.println(i+3);

}

}

public class Test{

public static void main (String[] args){

new B(5);

}

}

Options are

A.A B 8

B.A 5 B 8

C.A B 5

D.B 8 A 5

Answer :A is the correct answer.

Explanation:Constructor of class B call their superclass constructor of class A (public A()) , which execute first, and that constructors can be overloaded. Then come to constructor of class

B (public B (int i)).

29)What is the output for the below code ?

1. public interface InfA {

2. protected String getName();

3. }

public class Test implements InfA{

public String getName(){

return "test-name";

}

public static void main (String[] args){

Test t = new Test();

System.out.println(t.getName());

}

}

Options are

A.test-name

B.Compilation fails due to an error on lines 2

C.Compilation fails due to an error on lines 1

D.Compilation succeed but Runtime Exception

Answer :B is the correct answer.

Explanation:Illegal modifier for the interface method InfA.getName(); only public and abstract are permitted

30)What is the output for the below code ?

public class D {

int i;

int j;

public D(int i,int j){

this.i=i;

this.j=j;

}

public void printName() {

System.out.println("Name-D");

}

}

1. public class Test{

2. public static void main (String[] args){

3. D d = new D();

4. d.printName();

5.

6. }

7. }

Options are

A.Name-D

B.Compilation fails due to an error on lines 3

C.Compilation fails due to an error on lines 4

D.Compilation succeed but no output

Answer :B is the correct answer.

Explanation:Since there is already a constructor in this class (public D(int i,int j)), the compiler won't supply a default constructor. If you want a no-argument constructor to overload the witharguments version you already have, you have to define it by yourself. The constructor

D() is undefined in class D. If you define explicit constructor then default constructor will not be available. You have to define explicitly like public D(){ } then the above code will work. If no constructor into your class , a default constructor will be automatically generated by the compiler.