1 What is the result of compiling and running the following code?

public class Tester

{

static void test(int[] a)

{

int[] b = new int[2];

a = b;

System.out.print(b.length);

System.out.print(a.length);

}

public static void main(String[] args)

{

int[] a = new int[5];

test(a) ;

System.out.print(a.length);

}}

1. **225**
2. 255
3. 200

2 What is true after running the following code?

public static void main(String[] args)

{

String entries[] = {"entry1","entry2"};

Int count=0;

while (entries [count++]!=null)

{

System.out.println(count);

}

System.out.println(count);

}

1. **An Exception will be thrown**
2. 0 will be printed as part of the output
3. 2 will be printed as part of the output
4. 3 will be printed as part of the output

3 What is the result of compiling and running the following code?

public static void main(String[] args) {

int[] a = new int[0];

System.out.print(a.length);

}

1. **0**
2. Compilation error, arrays cannot be initialized to zero size
3. Compilation error, it is a.length() not a.length
4. None of Above

5 Which of the following declarations of an array is incorrect?

public static void main(String[] args) {

int[] a[];

int b[3];

int []c[];

int[] d[];

}

1. a
2. **b**
3. c
4. d

6 which of the followings are correct when tried to compile and execute the below code? (Choose 2)

class s

{

public static void main(String agr[])

{

short s1=4; //LINE 1

short s2 = s1+=s1; //LINE 2

short s3= s1+s2; //LINE 3

byte b1=(byte)s1 +(byte)s2; //LINE 4

byte b2=(byte)((byte)s1 +(byte)(byte)s2); //LINE 5

}}

1. compile time error at LINE 1
2. compile time error at LINE 2
3. **compile time error at LINE 3 and 4**
4. compile time error at LINE 5
5. compiles successfully.

Note : In Java, the default numeric type is integer. Hence when you add up two short type or 1 short type with a numeric constants then the result get converted to premitive integer type.   
  
Another concept to remember is that size of the numeric types grows in following order.  
byte-> short -> int ->double

short a = 2;  
  
short b = 3;  
  
short c = (short) (a + b);

7 What is the result of compiling and running the following code?

public class Tester

{

static void test(float x)

{

System.out.print("float");

}

static void test(double x) {

System.out.print("double");

}

public static void main(String[] args) {

test(99.9);

}}

1. float
2. **double**
3. Compilation error
4. Exception is thrown at run time.

8 What is the expected output?

interface Foldable {

public void fold() throws Exception ;

}

class Paper implements Foldable {

public void fold() { // line 6

System.out.print("Fold");

}}

public class Tester {

public static void main(String args []) {

Foldable obj1 = new Paper();

obj1.fold(); // line 8

Paper obj2 = new Paper(); // line 10

obj2.fold();

}

}

1. FoldFold
2. **Compilation error at line 6, fold() must declare at header throws Exception**
3. Compilation error at line 8, unhandled exception
4. RuntimeException at line 8
5. Compilation error at line 10, unhandled exception
6. RuntimeException at line 10

9 What is the result of compiling and running the following code?

public class Tester

{

public static void main(String[] args) {

System.out.print("1");

try {

System.out.print("2");

System.exit(0);

} finally {

System.out.print("3");

}

}

}

1. 123
2. **12**
3. Compilation error , there should be at least one catch before finally

10 The following code contains compilation errors, what of the following possible solutions (made together) can be done to remove these errors, putting in mind that all necessary imports will be added automatically? (choose two) (choose 2)

import java.io.File;

import java.text.DateFormat;

import java.util.Date;

public class Tester {

static void call() { // line 3

File file = new File("javachamp.dat"); //line 4

file.createNewFile(); // line 5

throw new IllegalArgumentException(); // line 6

}

public static void main(String[] args) {

call(); // line 11

DateFormat df = DateFormat.getDateInstance(); // line 12

Date parse = df.parse("12.11.2009"); // line 13

}

}

1. Add throws IOException declaration to call() at line 3
2. Add throws FileNotFoundException declaration to call() at line 3
3. Add throws IllegalArgumentException declaration to call() at line 3
4. Surround code from line 11 to 13 with try { } catch (IOException e) {} block
5. Surround code from line 11 to 13 with try { } catch (Exception e) {} block

11 In which case a NoClassDefFoundError will be thrown?

1. When try to compile a non found java file
2. When try to run a non found java .class file
3. When try to compile a non found java file or try to run a non found java .class file

12 What is the expected output of compiling and running the following code?

import java.io.IOException;

class AirPlane {

public AirPlane() throws IOException, RuntimeException {

System.out.println("AirPlane");

}

}

class AirJet extends AirPlane { } // line 7

public class Tester {

public static void main(String args[]) throws IOException { // line 10

new AirPlane(); // line 11

}

}

1. AirPlane
2. Compile error at line 7, AirJet must declare a constructor that throws IOException
3. or any of its supertypes
4. Compile error at line 10, main() must throw also RuntimeException
5. Compile error at line 11 ,new AirPlane() must be within try/catch block

13 What is the result of compiling and running the following code?

public class Tester {

public static void main(String[] args) {

String stmt = "javachamp 2009";

String[] arr = stmt.split(" ");

try {

int x = Integer.parseInt(arr[1]);

System.out.print(x);

} finally {

System.out.print("finally");

}

}

}

1. 2009
2. finally
3. 2009finally
4. No output will be produced
5. Compilation error

14 What is the output of the following code

public class Tester {

public static void main(String[] args) {

Double d = -4.0;

try {

d /= 0;

} catch (ArithmeticException e) {

System.out.println("EXCEPTION!");

} finally {

System.out.println(d);

} } }

1. NotANumber
2. NaN
3. EXCEPTION! -4.0
4. **-Infinity**
5. EXCEPTION! 0.0
6. Compilation fails

15 What is the result of compiling and running the following code?

public class Tester {

static void method(){

throw new Exception();

}

public static void main(String[] args) {

try {

method();

} catch (Throwable e) {

try {

throw new Exception() ;

} catch (Exception ex) {

System.out.print("exception");

} finally {

System.out.print("finally");

}

}

}

}

1. exception
2. finally
3. exceptionfinally
4. **Compilation error**

16 What is the result of compiling and running the following program?

public class Tester {

public static void main(String[] args) {

try {

throw new RuntimeException();

} catch (RuntimeException e) {

System.out.println("RuntimeException");

} catch (ArithmeticException e) {

System.out.println("ArithmeticException");

} catch (Exception e) {

System.out.println("Exception");

}

}

}

1. "RuntimeException" is printed
2. "Exception" is printed
3. "ArithmeticException" is printed
4. **Compilation error**

**Note**  : if we write runtime exception before it will give error that arithmetic exception is already exist

17 What is the result of compiling and running the following code?

public class Tester {

public static void main(String[] args) {

String stmt = "javachamp 2009";

String[] arr = stmt.split(" ");

try {

int x = Integer.parseInt(arr[0]);

System.out.print(x);

} catch (Exception e) {

System.out.print("catch");

} finally {

System.out.print("finally");

}

}

}

1. 2009
2. 2009finally
3. **catchfinally**
4. No output will be produced
5. Compilation error
6. 2009catchfinally

18 What is the expected output of compiling and running this code?

class Father {

public Father() throws RuntimeException {

System.out.print("Father");

throw new RuntimeException();

}

}

class Son extends Father {

public Son() throws RuntimeException {

System.out.print("Son");

}

}

public class Tester {

public static void main(String[] args) {

new Son(); // line 17

}

}

1. Compile error, an import to java.lang.RuntimeException is required
2. **Father will be printed then a RuntimeException would be thrown**
3. Compile error at line 17, new Son() should be within a try/catch block
4. Son
5. SonFather
6. FatherSon

19 What is the result of compiling and running the following code?

public class Tester {

public static void main(String[] args) {

System.out.print("1");

try {

return;

} catch (Exception e) {

System.out.print("2");

} finally {

System.out.print("3");

}

System.out.print("4");

}

}

1. 1234
2. **13**
3. 1
4. Compilation error

20 What is the result of compiling and running the following code?

public class Tester {

static void method() throws Exception {

throw new Exception();

}

public static void main(String[] args) {

try {

method();

} catch (Throwable e) {

try {

throw new Exception() ;

} catch (Exception ex) {

System.out.print("exception");

} finally {

System.out.print("finally");

}

}

}

}

1. "exception" is printed
2. "finally" is printed
3. "**exceptionfinally" is printed**
4. Compilation error

21 What is the expected result of compiling and running the following code?

import java.io.IOException;

class AirPlane

{

public AirPlane()

{

System.out.print("AirPlane");

}

}

class AirJet extends AirPlane {

public AirJet() throws IOException

{

try {

throw new IOException();

} catch (IOException e) {

System.out.print("IOException is thrown in AirJet");

}

}

}

public class Tester {

public static void main(String args[]) {

try {

new AirJet();

} catch (IOException e) {

System.out.print("IOException is thrown in Tester");

}

}

}

1. "AirPlaneIOException is thrown in AirJetIOException is thrown in Tester" will be printed
2. **"AirPlaneIOException is thrown in AirJet" will be printed**
3. "AirPlaneIOException is thrown in Tester" will be printed
4. Compilation error

Thread

22 What is the could be the output of compiling and running the following code? (choose two) (choose 2)

public class Test extends Thread {

static int count = 0;

public static void main(String argv[]) throws InterruptedException {

Test t = new Test ();

t.increment(count);

t.start();

Thread.sleep(1000);

System.out.println(count);

}

public void increment(int count) {

++count;

}

public void run() {

count = count + 5;

}

}

1. **5**
2. 6
3. Compilation error
4. An InterruptedException may be thrown

23 What is the result of compiling and running the following program?

public class Tester {

public void validate() {

int i = 0;

while (++i < 3) {

try {

wait();

} catch (InterruptedException e) {

e.printStackTrace();

}

System.out.print(i);

}

}

public static void main(String[] args) {

new Tester().validate();

}

}

1. Compilation error because of calling wait() outside a synchronized block
2. **Compilation error because IllegalMonitorStateException is not handled**
3. At runtime, it throws an IllegalMonitorStateException when trying to wait
4. 12

24 What is the possible result of compiling and running the following code?

public class Test implements Runnable {

Integer id;

public static void main(String[] args) {

new Thread(new Test()).start();

new Thread(new Test()).start();

}

public void run() {

press(id);

}

synchronized void press(Integer id) {

System.out.print(id.intValue());

System.out.print((++id).intValue());

}

}

1. 0101
2. 0011
3. -10-10
4. -1-100
5. **compilation error**
6. an exception is thrown at run time

25 When creating a thread by implementing Runnable interface :

1. you must implement the method run()
2. you can override run(), but in case you don't you'll be restricted to the provided run() method,which contains no code and does nothing

26 What is the output of compiling and running the following code?

class Writer extends Thread {

public void run() {

System.out.println("Writer run");

}

public static void main(String[] args) {

Runnable c = new Writer();

Thread t = new Thread(c);

t.run();

}

}

1. Compilation error
2. compiles fine, but no output produced
3. **compiles fine and prints "Writer run"**
4. compiles fine but throws an exception

27 What is the result of compiling and running the following code?

public class Tester extends Thread {

public void run() {

System.out.print("run");

}

public static void main(String[] args) {

Tester thread = new Tester();

new Thread(thread).start();

new Thread(thread).start();

}

}

1. Compilation error, can't invoke start() twice
2. runrun
3. IllegalThreadStateException will be thrown because of the second invoke to start()
4. Run

28 A thread that invokes the wait() method of an object, must owns the lock of the object.

1. True
2. False

29 What is the possible result of compiling and running the following code?

class Swimmer implements Runnable {

String name;

Pool pool;

Swimmer(String name, Pool pool) {

this.name = name;

this.pool = pool;

}

public void run() {

pool.swimIn(name);

}

}

public class Pool {

public void swimIn(String name) {

synchronized {

System.out.print(name);

System.out.print(name);

}

}

public static void main(String[] args) {

Pool pool = new Pool();

new Thread(new Swimmer("Tom", pool)).start();

new Thread(new Swimmer("Hanks", pool)).start();

}

}

1. TomTomHanksHanks
2. HanksHanksTomTom
3. HanksTomHanksTom
4. undetermined order
5. **compilation error // synchronized placed wrongly**

30 What is ture?

public class Test implements Runnable {

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

Test test = new Test();

Thread t= new Thread(test);

t.start();

t.join();

System.out.print("main");

}

public void run() {

System.out.print("run");

}

}

1. the output could be "mainrun"
2. **the output could be "runmain"**
3. the output could be "run" then an exception is thrown at run time
4. compilation error

31 What is considered an impossible output of running the following program?

public class Tester extends Thread {

int code = 9;

public void run() {

this.code = 7;

}

public static void main(String[] args) {

Tester thread = new Tester();

thread.start();

for (int i = 0; i < 5; i++) {

System.out.print(thread.code);

} } }

1. 99777
2. **97777**
3. 77777
4. 79999
5. 99999

32 How many times the statement "we are painting" would be printed in this program?

public class Test{

public static void main(String[] args) {

Painter painter1 = new Painter();

painter1.start();

Painter painter2 = new Painter();

painter2.start();

}

}

class Painter implements Runnable {

public void run() {

System.out.println("we are painting");

}

}

1. two times
2. zero times
3. **the program will not compile // Runnable does not have start method**

33 What is the output of running the following program?

class Tester extends Thread {

int total;

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

Tester t = new Tester();

t.start();

System.out.println("hi how are you:");

synchronized (t) {

System.out.println("waiting for t to complete");

t.wait();

System.out.println("total" + t.total);

}

}

synchronized public void run() {

for (int i = 0; i < 3; i++) {

total = total + i;

}

}

}

1. **main thread will wait indefinitely.**
2. total 0 is part of the output.
3. total 3 is part of the output .
4. it depends upon which thread has got the cpu first .

34 The idea of this program is to allow two workers to build a wall (which consists of bricks and cement) , given the following code, what necessarily modifications are needed to build a wall by alternating between bricks and cement (brickcementbrickcement...) and to avoid as possible, one worker monopolizing the work alone ? (choose all what apply) (choose 2)

class Worker extends Thread {

Contract contract;

Worker(Contract contract) {

this.contract = contract;

}

public void run() {

contract.work();

}

}

public class Contract {

StringBuilder wall = new StringBuilder("brick");

boolean isCementLastAdded = false;

public void putBrick() {

if (isCementLastAdded && !isWallDone()) {

wall.append("brick");

isCementLastAdded = false;

}

}

public void putCementLayer() {

if (!isCementLastAdded && !isWallDone()) {

wall.append("cement");

isCementLastAdded = true;

}

}

public boolean isWallDone() {

return wall.length() >= 100;

}

public void work() {

while (!isWallDone()) {

putCementLayer();

putBrick();

}

}

public static void main(String[] args) {

Contract contract = new Contract();

new Worker(contract).start();

new Worker(contract).start();

}

}

1. synchronize putBrick()
2. synchronize putCementLayer()
3. synchronize work()
4. add Thread.sleep() after putBrick() in work() method
5. synchronize isWallDone()

36 What is the possible output of compiling and running the following code? (choose 2)

public class Test {

public static void main(String[] args) {

Thread request1 = new Thread(new InternetRequest (),"request#1 ");

Thread request2 = new Thread(new InternetRequest (),"request#2 ");

request1.start();

request2.start();

} }

class InternetRequest implements Runnable {

public void run() {

System.out.print(Thread.currentThread().getName());

}}

1. request#2 request#1
2. **request#1 request#2**
3. an InterruptedException is thrown
4. request#1 request#1

37 Which of the following is a correct instantiation for a BufferedWriter? (choose 2)

1. BufferedWriter bw = new BufferedWriter(new File("data.txt"));
2. **BufferedWriter bw = new BufferedWriter(new FileWriter("data.txt"));**
3. **BufferedWriter bw = new BufferedWriter(new PrintWriter("data.txt"));**
4. BufferedWriter bw = new BufferedWriter("data.txt");

38 Which of the following statements is a In correct instantiation of PrintWriter?

1. PrintWriter writer = new PrintWriter("file.dat");
2. **PrintWriter writer = new PrintWriter(new BufferedWriter("file.dat"));**
3. PrintWriter writer = new PrintWriter(new FileWriter("file.dat"));
4. PrintWriter writer = new PrintWriter(new File("file.dat"));

39 At what line in the following program the file "data.txt" will be created in the file system?

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

public class Tester {

public static void main(String[] args) {

try {

File file = new File("data.txt");// line 5

file.createNewFile(); // line 6

FileWriter fr = new FileWriter(file); // line 7

BufferedWriter br = new BufferedWriter(fr); // line 8

br.append("javachamp");

br.flush();

br.close(); // line 11

} catch (IOException e) {

e.printStackTrace();

}

}

}

1. Line 5
2. **Line 6**
3. Line 7
4. Line 8
5. Line 11

40 Given the following code segment enclosed within a try/catch block, what valid Exception type can be catched causing no compilation error? (choose three) (choose 3)

try {

File file = new File("file.dat");

file.createNewFile();

} catch (

// INSERT EXCEPTION TYPE

e) {

e.printStackTrace();

}

1. **Exception**
2. RuntimeException
3. **Throwable**
4. FileNotFoundException
5. **IOException**

Interface and abstract classes

41 Will the following code compile correctly?

abstract class AirPlane {

abstract void fly();

void land() { // line 5

System.out.print("Landing..");

}

}

class AirJet extends AirPlane {

AirJet() {

super(); // line 14

}

void fly() {

System.out.print("Flying..");

}

}

1. Yes, it will compile with no errors
2. No, because at line 5 method land() must be abstract since class AirPlane is abstract
3. No, because class AirJet must override method land()
4. **No, because at line 14 AirJet constructor is calling the super() while AirPlane has no constructor defined**

42 To resolve the compilation error(s) in the following code, what can be done independently (choose two) : (choose 2)

interface Movable {

public abstract void m1(); // line 1

void m2(); // line 2

public void m3(); // line 3

abstract void m4(); // line 4

}

class Chair implements Movable { // line 5

public void m1() { } // line 6

void m2() { } // line 7

}

1. mark class Chair "abstract"
2. **mark Chair "abstract" and mark m2() "public"**
3. implement m3() and m4() in Chair (with public access modifier)
4. implement the methods m3() and m4() in Chair (with public access modifier) and mark m2() in Chair "public"

43 What is the result of compiling and running the following code?

import java.util.SortedMap;

import java.util.TreeMap;

public class Test {

public static void main(String[] args) {

TreeMap<Integer,String> map = new TreeMap<Integer,String>();

map.put(1, "one");

map.put(2, "two");

map.put(3, "three");

map.put(4, "four");

SortedMap<Integer, String> smap1 = map.tailMap(2);

SortedMap<Integer, String> smap2 = smap1.headMap(4);

SortedMap<Integer, String> smap3 = smap2.subMap(2, 3);

System.out.println(smap3);

}

}

1. {2=two, 3=three, 4=four}
2. {2=two, 3=three}
3. **{2=two}**
4. no output is printed

44 What is the result of compiling and running the following code?

class Bird {

static {

System.out.print("static1 ");

}

{

System.out.print("init3 ");

}

public Bird() {

System.out.print("Bird ");

}

static {

System.out.print("static2 ");

}

}

public class Falcon extends Bird {

Falcon(){

System.out.print("Falcon ");

}

public static void main(String[] args) {

System.out.print("main ");

new Falcon();

}

}

1. main static1 static2 init3 Bird Falcon
2. main static1 static2 Falcon init3 Bird
3. main static1 static2 Falcon Bird init3
4. **static1 static2 main init3 Bird Falcon**
5. static1 static2 main Bird init3 Falcon

45 Which of the following statements produces the output 3 + 8 ? (choose 2)

public class Tester {

public static void main(String[] args) {

int x = 8;

int y = 3;

System.out.printf("%d + %d \n", y, x); // stmt1

System.out.printf("%f + %f \n", (float) y, (float) x);// stmt2

System.out.printf("%d + %d \n", x, y);// stmt3

System.out.format("%2$d + %1$d", x, y);// stmt4

}

}

1. **Statement 1**
2. Statement 2
3. Statement 3
4. **Statement 4**

46 Is the following declaration for interface Bendable correct and free of compilation error?

abstract interface Bendable { // line 1

final int x = 2009; // line 3

void method1(); // line 5

}

1. **Yes, this is a correct and free of error declaration**
2. No, compilation error at line 1, Bendable should be declared public abstract
3. No, compilation error at line 3 , x should be declared public final
4. No, compilation error at line 5 , method method1() should be declared public abstract

47 What is the result of compiling and running the following code?

enum Status {

Published(1), Rejected(2); // Line 2

int index; // Line 3

Status(int i) { index = i; }

int getIndex(){return index;} // Line 5

}

public class Tester {

public static void main(String [] args) {

System.out.println(Status.Published.index + Status.Rejected.index); // Line 10

}

}

1. Compilation error at line 2. Must be terminated by a colon ','
2. Compilation error at line 3. index must be marked static
3. Compilation error at line 5. must be marked public
4. Compilation error at line 10, can't access index directly, must use getIndex() instead
5. **3**

48 The following code contains one compilation error, where could it be?

public class Tester {

Tester() { } // line 1

static void Tester() { this(); } // line 2

public static void main(String[] args) { // line 3

Tester(); // line 4

}

}

1. At line 1, constructor Tester must be marked public like its class
2. **At line 2, constructor call “this()” can only be called inside constructors**
3. At line 3, compilation error, ambiguity problem, compiler can”t determine wether a constructor
4. Tester or method Tester is called

49 Given the following two classes (Address and Account), what may happen if you attempt to serialize an instance of Account?

1.class Address{}

2.class Account implements Serializable {Address address;}

1. Compilation error
2. Exception is thrown at run time
3. **Serialization will succeed correctly with no problems**

50 what will be the output when we compile and execute the code below:

class Compound{

public static void main(String args[])

{

int k=12;

k/=--k;

System.out.println(k);

k\*=k++;

System.out.println(k);

k\*=++k;

System.out.println(k);

}

}

1. It will compile successfully and display 1 followed by 2 and then 3 as an output.
2. **It will compile successfully and display 1 followed by 1 and then 2 as an output.**
3. It will compile successfully and display 1 followed by 3 and then 4 as an output.
4. It will generate compile time error.

51 An object is subject to garbage collection in java :

1. when the program is at a point of execution that is out the scope of the object
2. when the object is set to null
3. **when the object becomes unreachable**

**52**  What will be the output :

class MySuper

{

MySuper(){disp();}

void disp(){System.out.println("superclass");}

}

class MySub extends MySuper

{

double i=Math.ceil(8.4f);

public static void main(String arg[])

{

MySuper obj= new MySub();

obj.disp();

}

void disp(){System.out.println(i);}

}

1 The program displays "superclass" followed by "9.0" as an output.

2 The program displays "superclass" followed by "superclass" as an output.

3 The program displays "9.0" followed by "9.0" as an output.

4 **The program displays "0.0" followed by "9.0" as an output.**