1.

What will be the output of the following class:

public class TestClass

{

public void testRefs(String str, StringBuffer sb)

{

str = str + sb.toString();

sb.append(str);

str = null;

sb = null;

}

public static void main(String[] args)

{

String s = "aaa";

StringBuffer sb = new StringBuffer("bbb");

new TestClass().testRefs(s, sb);

System.out.println("s="+s+" sb="+sb);

}

}

Select 1 correct option.

a s=aaa sb=bbb

b s=null sb=null

c s=aaa sb=null

d s=null sb=bbbaaa

e s=aaa sb=bbbaaabbb

ans: e

2.

What will be the output of the following lines ?

System.out.println("" +5 + 6); //1

System.out.println(5 + "" +6); // 2

System.out.println(5 + 6 +""); // 3

System.out.println(5 + 6); // 4

Select 1 correct option.

a 56, 56, 11, 11

b 11, 56, 11, 11

c 56, 56, 56, 11

d 56, 56, 56, 56

e 56, 56, 11, 56

ans: a

3. What will be the result of attempting to compile the following program?

public class TestClass

{

long l1;

public void TestClass(long pLong) {l1 = pLong ; } //(1)

public static void main(String args[])

{

TestClass a, b ;

a = new TestClass(); //(2)

b = new TestClass(5); //(3)

}

}

a A compilation error will be encountered at (1), since constructors should not specify a return value.

b A compilation error will be encountered at (2), since the class does not have a default constructor.

c A compilation error will be encountered at (3).

d The program will compile correctly.

e It will not compile because parameter type of the constructor is different than the type of value passed to it.

Ans: c

4.

Carefully examine the following code.

public class StaticTest

{

static

{

System.out.println("In static");

}

{

System.out.println("In non - static");

}

public static void main(String args[ ])

{

StaticTest st1; //1

System.out.println(" 1 ");

st1 = new StaticTest(); //2

System.out.println(" 2 ");

StaticTest st2 = new StaticTest(); //3

}

}

What will be the output?

Select 1 correct option.

a In static, 1, In non - static, 2, In non - static : in that order.

The static block will be executed only once when the class is loaded. A class is loaded when it is first referenced at line 1.

b Compilation error.

c 1, In static, In non - static, 2, In non - static : in that order.

d In static, 1, In non - static, 2, In non - static : in unknown order.

e None of the above.

Ans; a

5.

What will the following code snippet print?

int index = 1;

String[] strArr = new String[5];

String myStr = strArr[index];

System.out.println(myStr);

Select 1 correct option.

A It will print nothing.

B It will print 'null'

C It will throw ArrayIndexOutOfBounds at runtime.

D It will print NullPointerException at runtime.

E None of the above.

Ans : B

6.

What will the following program print?

public class TestClass

{

public static void main(String[] args)

{

unsigned byte b = 0;

b--;

System.out.println(b);

}

}

Select 1 correct option.

A 0

B -1

C 255

D -128

E It will not compile.

ANS : E

7.

What will happen when you compile and run the following program using the command line:

java TestClass 1 2

public class TestClass

{

public static void main(String[] args)

{

int i = Integer.parseInt(args[1]);

System.out.println(args[i]);

}

}

Select 1 correct option.

a It'll print 1

b It'll print 2

c It'll print some junk value.

d It'll throw ArrayIndexOutOfBoundsException

e It'll throw NumberFormatException

ANS : D

8.

Which of the following are correct ways to initialize the static variables MAX and CLASS\_GUID ?

class Widget

{

static int MAX; //1

static final String CLASS\_GUID; // 2

Widget()

{

//3

}

Widget(int k)

{

//4

}

}

Select 2 correct options

a Modify lines //1 and //2 as : static int MAX = 111; static final String CLASS\_GUID = "XYZ123";

b Add the following line just after //2 : static { MAX = 111; CLASS\_GUID = "XYZ123"; }

c Add the following line just before //1 : { MAX = 111; CLASS\_GUID = "XYZ123"; }

d Add the following line at //3 as well as //4 : MAX = 111; CLASS\_GUID = "XYZ123";

e Only option 3 is valid.

ANS : A,b

9.

Consider the following code snippet ...

boolean[] b1 = new boolean[2];

boolean[] b2 = {true , false};

System.out.println( "" + (b1[0] == b2[0]) + ", "+ (b1[1] == b2[1]) );

What will it print ?

Select 1 correct option.

a It will not compile.

b It will throw ArrayIndexOutOfBoundsError at Runtime.

c It will print false, true.

d It will print true, false.

e It will print false, false.

ANS : c

10.

Which of the following are valid identifiers?

Select 2 correct options

a class

b $value$

c angstrom

d 2much

e zer@

ANS : B,c

11.

public class TestClass

{

public static void main(String[] args)

{

String str = "111";

boolean[] bA = new boolean[1];

if( bA[0] ) str = "222";

System.out.println(str);

}

}

What will the above program print?

Select 1 correct option.

a 111

b 222

c It will not compile as bA[0] is uninitialized.

d It will throw an exception at runtime.

e None of the above.

ANS: A

12.

Which of the following is not a primitive data value in Java?

Select 2 correct options

a "x"

b 'x'

c 10.2F

d Object

e false

ANS : A,d

13.

What will the following program print?

public class TestClass

{

static boolean b;

static int[] ia = new int[1];

static char ch;

static boolean[] ba = new boolean[1];

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

{

boolean x = false;

if( b )

{

x = ( ch == ia[ch]);

}

else x = ( ba[ch] = b );

System.out.println(x+" "+ba[ch]);

}

}

Select 1 correct option.

a true true

b true false

c false true

d false false

e It'll not compile.

ANS : D

14.

Which code fragments will print the last argument given on the command line to the standard output, and exit without any output and exceptions if no arguments are given?

1.

public static void main(String args[ ])

{

if (args.length != 0) System.out.println(args[args.length-1]);

}

2.

public static void main(String args[ ])

{

try { System.out.println(args[args.length-1]); }

catch (ArrayIndexOutOfBoundsException e) { }

}

3.

public static void main(String args[ ])

{

int i = args.length;

if (i != 0) System.out.println(args[i-1]);

}

4.

public static void main(String args[ ])

{

int i = **args.length-1;**

if (i > 0) System.out.println(args[i]);

}

5.

public static void main(String args[ ])

{

try { System.out.println(args[args.length-1]); }

catch (NullPointerException e) {}

}

Select 3 correct options

a Code No. 1

b Code No. 2

c Code No. 3

d Code No. 4

e Code No. 5

ANS : A,b,c

15.

Given the following class, which of these given blocks can be inserted at line 1 without errors?

public class InitClass

{

private static int loop = 15 ;

static final int INTERVAL = 10 ;

boolean flag ;

//line 1

}

Select 4 correct options

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

b static { loop = 1; }

c static { loop += INTERVAL; }

d static { INTERVAL = 10; }

e { flag = true; loop = 0; }

ANS : a,b,c,e