

MCQ (Core Java)

```
1) void start() {  
    A a = new A();  
    B b = new B();  
    a.s(b);  
    b = null; /* Line 5 */  
    a = null; /* Line 6 */  
    System.out.println("start completed"); /* Line 7 */  
}
```

When is the B object, created in line 3, eligible for garbage collection?

a. after line 5

b. after line 6

c. after line 7

d. There is no way to be absolutely certain.

```
2) class HappyGarbage01  
{  
    public static void main(String args[])  
    {  
        HappyGarbage01 h = new HappyGarbage01();  
        h.methodA(); /* Line 6 */  
    }  
    Object methodA()  
    {  
        Object obj1 = new Object();  
        Object [] obj2 = new Object[1];  
        obj2[0] = obj1;  
        obj1 = null;  
        return obj2[0];  
    }  
}
```

```
}  
}
```

Where will be the most chance of the garbage collector being invoked?

- a) After line 9
- b) After line 10
- c) After line 11
- d) Garbage collector never invoked in methodA()

3) class Bar { }

class Test

```
{  
    Bar doBar()  
    {  
        Bar b = new Bar(); /* Line 6 */  
        return b; /* Line 7 */  
    }  
    public static void main (String args[])  
    {  
        Test t = new Test(); /* Line 11 */  
        Bar newBar = t.doBar(); /* Line 12 */  
        System.out.println("newBar");  
        newBar = new Bar(); /* Line 14 */  
        System.out.println("finishing"); /* Line 15 */  
    }  
}
```

At what point is the Bar object, created on line 6, eligible for garbage collection?

- a) after line 12
- b) after line 14
- c) after line 7, when doBar() completes
- d) after line 15, when main() completes

4) class Test

```
{  
    private Demo d;  
    void start()  
    {  
        d = new Demo();  
        this.takeDemo(d); /* Line 7 */  
    } /* Line 8 */  
    void takeDemo(Demo demo)  
    {  
        demo = null;  
        demo = new Demo();  
    }  
}
```

When is the Demo object eligible for garbage collection?

- a) After line 7
- b) After line 8
- c) After the start() method completes
- d) When the instance running this code is made eligible for garbage collection.

5) public class X

```
{  
    public static void main(String [] args)  
    {  
        X x = new X();  
        X x2 = m1(x); /* Line 6 */  
        X x4 = new X();  
        x2 = x4; /* Line 8 */  
    }  
}
```

```

        doComplexStuff();
    }
    static X m1(X mx)
    {
        mx = new X();
        return mx;
    }
}

```

After line 8 runs. how many objects are eligible for garbage collection?

- a) 0
- b) 1**
- c) 2
- d) 3

6)

```

public Object m()
{
    Object o = new Float(3.14F);
    Object [] oa = new Object[1];
    oa[0] = o; /* Line 5 */
    o = null; /* Line 6 */
    oa[0] = null; /* Line 7 */
    return o; /* Line 8 */
}

```

When is the Float object, created in line 3, eligible for garbage collection?

- a) just after line 5
- b) just after line 6
- c) just after line 7**
- d) just after line 8

7.

```
class X2
{
    public X2 x;
    public static void main(String [] args)
    {
        X2 x2 = new X2(); /* Line 6 */
        X2 x3 = new X2(); /* Line 7 */
        x2.x = x3;
        x3.x = x2;
        x2 = new X2();
        x3 = x2; /* Line 11 */
        doComplexStuff();
    }
}
```

after line 11 runs, how many objects are eligible for garbage collection?

- a) 0
- b) 1
- c) 2**
- d) 3

8) What allows the programmer to destroy an object x?

- a) x.delete()
- b) x.finalize()
- c) Runtime.getRuntime().gc()
- d) Only the garbage collection system can destroy an object.

9) public class Outer

```
{
    public void someOuterMethod()
    {
```

```

        //Line 5
    }
    public class Inner { }

    public static void main(String[] argv)
    {
        Outer ot = new Outer();
        //Line 10
    }
}

```

Which of the following code fragments inserted, will allow to compile?

- a) new Inner(); //At line 5
- b) new Inner(); //At line 10
- c) new ot.Inner(); //At line 10
- d) new Outer.Inner(); //At line 10

10) interface Base

```

{
    boolean m1 ();
    byte m2(short s);
}

```

which two code fragments will compile?

```

interface Base2 implements Base {}
abstract class Class2 extends Base
{ public boolean m1(){ return true; }}
abstract class Class2 implements Base {}
abstract class Class2 implements Base
{ public boolean m1(){ return (7 > 4); }}
abstract class Class2 implements Base
{ protected boolean m1(){ return (5 > 7) }}

```

- a) 1 and 2

- b) 2 and 3
- c) 3 and 4
- d) 1 and 5

11)

Which three form part of correct array declarations?

`public int a []`

`static int [] a`

`public [] int a`

`private int a [3]`

`private int [3] a []`

`public final int [] a`

- a) 1, 3, 4
- b) 2, 4, 5
- c) 1, 2, 6
- d) 2, 5, 6

12)

`public class Test { }`

What is the prototype of the default constructor?

- a) `Test()`
- b) `Test(void)`
- c) `public Test()`
- d) `public Test(void)`

13) Which cause a compiler error?

- a) `int[] scores = {3, 5, 7};`
- b) `int [][] scores = {2,7,6}, {9,3,45};`
- c) `String cats[] = {"Fluffy", "Spot", "Zeus"};`
- d) `boolean results[] = new boolean [] {true, false, true};`
- e) `Integer results[] = {new Integer(3), new Integer(5), new Integer(8)};`

14)

Which three are valid method signatures in an interface?

`private int getArea();`

```
public float getVol(float x);  
public void main(String [] args);  
public static void main(String [] args);  
boolean setFlag(Boolean [] test);
```

- a) 1 and 2
- b) 2, 3 and 5**
- c) 3, 4, and 5
- d) 2 and 4

15) What is the widest valid returnType for methodA in line 3?

```
public class ReturnIt  
{  
    returnType methodA(byte x, double y) /* Line 3 */  
    {  
        return (long)x / y * 2;  
    }  
}
```

- a) int
- b) byte
- c) long
- d) double**

16) class A

```
{  
    protected int method1(int a, int b)  
    {  
        return 0;  
    }  
}
```

Which is valid in a class that extends class A?

- a) `public int method1(int a, int b) {return 0; }`
- b) `private int method1(int a, int b) { return 0; }`
- c) `public short method1(int a, int b) { return 0; }`
- d) `static protected int method1(int a, int b) { return 0; }`

17) /* Missing Statement ? */

```
public class foo
{
    public static void main(String[]args)throws Exception
    {
        java.io.PrintWriter out = new java.io.PrintWriter();
        new java.io.OutputStreamWriter(System.out,true);
        out.println("Hello");
    }
}
```

What line of code should replace the missing statement to make this program compile?

- a) No statement required.
- b) `import java.io.*;`
- c) `include java.io.*;`
- d) `import java.io.PrintWriter;`

18) What will be the output of the program?

```
public class Foo
{
    public static void main(String[] args)
    {
        try
        {
            return;
        }
        finally
        {
```

```

        System.out.println( "Finally" );
    }
}
}

```

- a) **Finally**
- b) Compilation fails.
- c) The code runs with no output.
- d) An exception is thrown at runtime.

19)

What will be the output of the program?

```

try
{
    int x = 0;
    int y = 5 / x;
}
catch (Exception e)
{
    System.out.println("Exception");
}
catch (ArithmeticException ae)
{
    System.out.println(" Arithmetic Exception");
}
System.out.println("finished");

```

- a) finished
- b) Exception
- c) **Compilation fails.**
- d) Arithmetic Exception

20)

What will be the output of the program?

```

public class X
{
    public static void main(String [] args)
    {
        try
        {
            badMethod();
            System.out.print("A");
        }
        catch (Exception ex)
        {
            System.out.print("B");
        }
        finally
        {
            System.out.print("C");
        }
        System.out.print("D");
    }
    public static void badMethod()
    {
        throw new Error(); /* Line 22 */
    }
}

```

- a) ABCD
- b) Compilation fails.
- c) C is printed before exiting with an error message.
- d) BC is printed before exiting with an error message.

21)

What will be the output of the program?

```
public class X
{
    public static void main(String [] args)
    {
        try
        {
            badMethod();
            System.out.print("A");
        }
        catch (RuntimeException ex) /* Line 10 */
        {
            System.out.print("B");
        }
        catch (Exception ex1)
        {
            System.out.print("C");
        }
        finally
        {
            System.out.print("D");
        }
        System.out.print("E");
    }
    public static void badMethod()
    {
        throw new RuntimeException();
    }
}
```

- a) BD
- b) BCD
- c) BDE**
- d) BCDE

22)

What will be the output of the program?

```
public class RTEExcept
{
    public static void throwit ()
    {
        System.out.print("throwit ");
        throw new RuntimeException();
    }
    public static void main(String [] args)
    {
        try
        {
            System.out.print("hello ");
            throwit();
        }
        catch (Exception re )
        {
            System.out.print("caught ");
        }
        finally
        {
            System.out.print("finally ");
        }
        System.out.println("after ");
    }
}
```

- a) hello throwit caught
- b) Compilation fails
- c) hello throwit RuntimeException caught after
- d) hello throwit caught finally after

23) public class Myfile

```
{
    public static void main (String[] args)
    {
        String biz = args[1];
        String baz = args[2];
        String rip = args[3];
        System.out.println("Arg is " + rip);
    }
}
```

Select how you would start the program to cause it to print: Arg is 2

- a) java Myfile 222
- b) java Myfile 1 2 2 3 4
- c) java Myfile 1 3 2 2
- d) java Myfile 0 1 2 3

24)

```
public void foo( boolean a, boolean b)
{
    if( a )
    {
        System.out.println("A"); /* Line 5 */
    }
    else if(a && b) /* Line 7 */
    {
        System.out.println( "A && B");
    }
}
```

```

else /* Line 11 */
{
    if ( !b )
    {
        System.out.println( "notB" );
    }
    else
    {
        System.out.println( "ELSE" );
    }
}
}

```

- a) If a is true and b is true then the output is "A && B"
- b) If a is true and b is false then the output is "notB"
- c) If a is false and b is true then the output is "ELSE"
- d) If a is false and b is false then the output is "ELSE"

25)

```

switch(x)
{
    default:
        System.out.println("Hello");
}

```

Which two are acceptable types for x?

byte

long

char

float

Short

Long

- a) 1 and 3
- b) 2 and 4
- c) 3 and 5
- d) 4 and 6

26)

```
public void test(int x)
{
    int odd = 1;
    if(odd) /* Line 4 */
    {
        System.out.println("odd");
    }
    else
    {
        System.out.println("even");
    }
}
```

Which statement is true?

- a) Compilation fails.
- b) "odd" will always be output.
- c) "even" will always be output.
- d) "odd" will be output for odd values of x, and "even" for even values.

27)

```
public class While
{
    public void loop()
    {
        int x= 0;
        while ( 1 ) /* Line 6 */
        {
            System.out.print("x plus one is " + (x + 1)); /* Line 8 */
        }
    }
}
```



```
    }  
  }  
}
```

Which statement is true?

- a) There is a syntax error on line 1.
- b) There are syntax errors on lines 1 and 6.
- c) There are syntax errors on lines 1, 6, and 8.
- d) There is a syntax error on line 6.**

28)

```
public void foo( boolean a, boolean b)  
{  
    if( a )  
    {  
        System.out.println("A"); /* Line 5 */  
    }  
    else if(a && b) /* Line 7 */  
    {  
        System.out.println( "A && B");  
    }  
    else /* Line 11 */  
    {  
        if ( !b )  
        {  
            System.out.println( "notB" );  
        }  
        else  
        {  
            System.out.println( "ELSE" );  
        }  
    }  
}
```

}

- a) If a is true and b is true then the output is "A && B"
- b) If a is true and b is false then the output is "notB"
- c) If a is false and b is true then the output is "ELSE"
- d) If a is false and b is false then the output is "ELSE"

29)

switch(x)

{

default:

System.out.println("Hello");

}

Which two are acceptable types for x?

byte

long

char

float

Short

Long

- a) 1 and 3
- b) 2 and 4
- c) 3 and 5
- d) 4 and 6

A

30)

public void test(int x)

{

int odd = 1;

if(odd) /* Line 4 */

{

System.out.println("odd");

}

```
else
{
    System.out.println("even");
}
}
```

Which statement is true?

- a) **Compilation fails.**
- b) "odd" will always be output.
- c) "even" will always be output.
- d) "odd" will be output for odd values of x, and "even" for even values.

31)

```
public class While
{
    public void loop()
    {
        int x= 0;
        while ( 1 ) /* Line 6 */
        {
            System.out.print("x plus one is " + (x + 1)); /* Line 8 */
        }
    }
}
```

Which statement is true?

- a) There is a syntax error on line 1.
- b) There are syntax errors on lines 1 and 6.
- c) There are syntax errors on lines 1, 6, and 8.
- d) **There is a syntax error on line 6.**