1.

What will the following program print?

```
public class TestClass
{
  static int someInt = 10;
  public static void changeIt(int a)
  {
    a = 20;
  }
  public static void main(String[] args)
  {
    changeIt(someInt);
    System.out.println(someInt);
  }
}
```

See Hint

Please select 1 option

- ○ 10
- ○ 20
- ○ It will not compile.
- ○ It will throw an exception at runtime.
- ○ None of the above.

2.

Under what situations does a class get a default constructor?

Please select 1 option

- ○ All classes in Java get a default constructor.
- ○ You have to define at least one constructor to get the default constructor.
- ○ If the class does not define any constructors explicitly.
- ○ All classes get default constructor from Object class.
- ○ None of the above.

3.

You are writing a class that represents the equation of a straight line:
`y = mx + c;`
This class has only one method named `calcY` that takes the value of `x` and returns the value of `y`.
Which variable scopes will you use to store the values of `m` and `c` in an instance of this class?

Please select 1 option

- ○ global variables
- ○ static variables
- ○ instance variables
- ○ local variables

4.

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
   }
}
```

Please select 1 option

- ○ A compilation error will be encountered at //1, since constructors should not specify a return value.
- ○ A compilation error will be encountered at //2, since the class does not have a default constructor.
- ○ A compilation error will be encountered at //3.
- ○ The program will compile correctly.
- ○ It will not compile because parameter type of the constructor is different than the type of value passed to it.

5.

What will the following class print when compiled and run?

```
class Holder
{
   int value = 1;
   Holder link;
   public Holder(int val){ this.value = val; }
   public static void main(String[] args)
   {
      final Holder a = new Holder(5);
      Holder b = new Holder(10);
      a.link = b;
      b.link = setIt(a, b);
      System.out.println(a.link.value+" "+b.link.value);
   }

   public static Holder setIt(final Holder x, final Holder y)
   {
       x.link = y.link;
       return x;
   }

}
```

Please select 1 option

- ○ It will not compile because 'a' is final.
- ○ It will not compile because method setIt() cannot change x.link.
- ○ It will print 5, 10.
- ○ It will print 10, 10.
- ○ It will throw an exception when run.

6.

What would be the result of trying to compile and run the following program?

```
public class Test
{
   int[] ia = new int[1];
   Object oA[]  = new Object[1];
   boolean bool;
   public static void main(String args[])
   {
      Test test = new Test();
      System.out.println(test.ia[0] + "  " + test.oA[0]+"  "+test.bool);
   }
}
```

Please select 1 option

- ○ The program will fail to compile, because of uninitialized variable 'bool'.
- ○ The program will throw a java.lang.NullPointerException when run.
- ○ The program will print "0 null false".
- ○ The program will print "0 null true".
- ○ The program will print null and false but will print junk value for ia[0].

7.

Which lines contain a valid constructor in the following code?

```
public class TestClass
{
   public TestClass(int a, int b) { } // 1
   public void TestClass(int a) { }   // 2
   public TestClass(String s); // 3
   private TestClass(String s, int a) { }     //4
   public TestClass(String s1, String s2) { }; //5
}
```

Please select 3 options

- ☐ Line // 1
- ☐ Line // 2
- ☐ Line // 3
- ☐ Line // 4
- ☐ Line // 5

8.

Which of the following are true about the "default" constructor?

Please select 2 options

- ☐ It is provided by the compiler only if the class does not define any constructor.
- ☐ It initializes the instance members of the class.
- ☐ It calls the default 'no-args' constructor of the super class.
- ☐ It initializes instance as well as class fields of the class.
- ☐ It is provided by the compiler if the class does not define a 'no- args' constructor.

9.

Given:

```
class Acct {
    int id;
    double balance;

    public Acct(int id, double balance){
        this.id = id;
        this.balance =  balance;
    }
    public void setId(int id){
        this.id = id;
    }
    public void setBalance(double balance){
        this.balance = balance;
    }
}
public class Account{
    public static void main(String[] args) {

        //INSERT CCODE HERE

        System.out.println(acct.id+" "+acct.balance);
    }
}
```

What can be inserted in the above code so that it will print `10 10.0`?

Please select 1 option

- ○
```
Acct acct = new Acct();
acct.id = 10;
acct.balance = 10.0;
```

- ○ `Acct acct = new Acct(10, 10);`

- ○
```
Acct acct = new Acct();
acct.setId(10);
acct.setBalance(10.0);
```

- ○
```
Acct acct = null;
acct.id = 10;
acct.balance = 10.0;
```

- ○
```
Acct acct;
acct.id = 10;
acct.balance = 10.0;
```

10.

Which of these statements regarding the following code are correct ?

```
public class TestClass
{
   static int a;
   int b;
   public TestClass()
   {
      int c;
      c = a;
      a++;
      b += c;
   }
   public static void main(String args[]) {   new TestClass();   }
}
```

Please select 1 option

- ○ The code will fail to compile, since the constructor is trying to access static members.
- ○ The code will fail to compile, since the constructor is trying to use static member variable a before it has been initialized.
- ○ The code will fail to compile, since the constructor is trying to use member variable b before it has been initialized.
- ○ The code will fail to compile, since the constructor is trying to use local variable c before it has been initialized.
- ○ The code will compile and run without any problems.

11.

Given the following code, which statements can be placed at the indicated position without causing compile and run time errors?

```
public class Test
{
   int i1;
   static int i2;
   public void method1()
   {
      int i;
      // ... insert statements here
   }
}
```

Please select 3 options

- ☐ `i = this.i1;`
- ☐ `i = this.i2;`
- ☐ `this = new Test( );`
- ☐ `this.i = 4;`
- ☐ `this.i1 = i2;`

12.

Which of the following can be used as a constructor for the class shell given below?

```
public class TestClass
{
   // lots of code ...
}
```

Please select 2 options

- ☐ `public void TestClass() {...}`
- ☐ `public TestClass() {...}`
- ☐ `public static TestClass() {...}`
- ☐ `public final TestClass() {...}`
- ☐ `public TestClass(int x) { ...}`

13.

Which of these statements are true?

Please select 2 options

- ☐ All classes must explicitly define a constructor.
- ☐ A constructor can be declared private.
- ☐ A constructor can declare a return value.
- ☐ A constructor must initialize all the member variables of a class.
- ☐ A constructor can access the non-static members of a class.

14.

What will happen when the following code is compiled?

```
public class FooBar
{
   private int FooBar;  //1
   public FooBar(int FooBar){ this.FooBar = FooBar; } //2
   public void FooBar(){ } //3
}
```

Please select 1 option

- ○ Compilation error at //1.
- ○ Compilation error at //2.
- ○ Compilation error at //3.
- ○ Compiles without any error.

15.

Given a class named Test, which of these would be valid definitions for the constructors for the class?

Please select 1 option

- ○ Test(Test b) { }
- ○ Test Test( ) { }
- ○ private final Test( ) { }
- ○ void Test( ) { }
- ○ public static void Test(String args[ ]) { }

16.

What will the following code print?

```
public class TestClass
{
        int x = 5;
        int getX(){ return x; }

        public static void main(String args[]) throws Exception
        {
            TestClass tc = new TestClass();
            tc.looper();
            System.out.println(tc.x);
        }

        public void looper()
        {
            int x = 0;
            while( (x = getX()) != 0 )
            {
                for(int m = 10; m>=0; m--)
                {
                    x = m;
                }
            }

        }

}
```

See Hint

Please select 1 option

- ○ It will not compile.
- ○ It will throw an exception at runtime.
- ○ It will print 0.
- ○ It will print 5.
- ○ None of these.

17.

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
   }
}
```

Please select 2 options

- [ ] Modify lines //1 and //2 as :
```
static int MAX = 111;
static final String CLASS_GUID = "XYZ123";
```

- [ ] Add the following line just after //2 :
```
static {  MAX = 111; CLASS_GUID = "XYZ123"; }
```

- [ ] Add the following line just before //1 :
```
{ MAX = 111; CLASS_GUID = "XYZ123"; }
```

- [ ] Add the following line at //3 as well as //4 :
```
MAX = 111; CLASS_GUID = "XYZ123";
```

- [ ] Only option 3 is valid.