```
1.  class A {
            int i;
            publicstaticvoid main(String[] args) {
                    System.out.println(i);
            }       }
```

What could be the output?

```
2 . class A {
            publicstaticvoid main(String[] args) {
                    byte b;
                    b = (b = 5) * 10;
                    System.out.println(b);
            }       }
3 . class A {
            A a1;
            publicstaticvoid main(String[] args) {
                    System.out.println(new A().a1);
            }       }

4.class Manager {
    publicstaticvoid main(String[] args) {
            boolean flag = true;
            if(flag = false) { System.out.println("Yes");
    }
            else {   System.out.println("No");        }
    }       }
5.class Test {
    Test() {System.out.println("Constructor");      }
    void Test() {     System.out.println("Method");  }
    publicstaticvoid main(String[] args){
            new Test().Test();
    }       }
 6. class X {
    X() {   System.out.println("1");          }
    static {  System.out.println("2");         }
    publicstaticvoid main(String[] args) {
            new X();
    System.out.println(“www.javapadho.com”);

    }       }
7. class A {
    void test() {
            System.out.println("Hello");
    }
}
class B extends A {
    voidanother() {
            System.out.println("Hi");
    }
}
class Manager {
    publicstaticvoid main(String[] args) {
            new B().another();
            A a1 = new B();
            a1.another();
    }
}
8. class A {
    A() {       this(25);
    System.out.println("Hello");
    }
    A(int i) {
            super();
            System.out.println("Hi");
    }
}
class Manager {
    public static void main(String[] args) {
            new A();
            new A(10);
    }
}
```

9)

Which statement about methods is true?

Select the one correct answer.

(a) A method is an implementation of an abstraction.

(b) A method is an attribute defining the property of a particular abstraction.

(c) A method is a category of objects.

(d) A method is an operation defining the behavior for a particular abstraction.

(e) A method is a blueprint for making operations.

10)

Given that Thing is a class, how many objects and how many reference variables are created by the following code?

```
Thing item, stuff;
item = new Thing();
Thing entity = new Thing();
```

Select the two correct answers.

(a) One object is created.

(b) Two objects are created.

(c) Three objects are created.

(d) One reference variable is created.

(e) Two reference variables are created.

(f) Three reference variables are created.

11) Which statement about instance members is true?

Select the one correct answer.

(a) An instance member is also called a static member.

(b) An instance member is always a field.

(c) An instance member is never a method.

(d) An instance member belongs to an instance, not to the class as a whole.

(e) An instance member always represents an operation.

12) What will be the result of compiling the following program?

```
public class MyClass {
 long var;
 public void MyClass(long param) { var = param; } // (1)
 public static void main(String[] args) {
 MyClass a, b;
 a = new MyClass(); // (2)
 b = new MyClass(5); // (3)
 }
}
```

Select the one correct answer.

(a) A compilation error will occur at (1), since constructors cannot specify a return value.

(b) A compilation error will occur at (2), since the class does not have a default
constructor.
(c) A compilation error will occur at (3), since the class does not have a constructor
that takes one argument of type int.
(d) The program will compile without errors.

13) Which one of the following array declaration statements is not legal?
Select the one correct answer.
(a) int []a[] = new int [4][4];
(b) int a[][] = new int [4][4];
(c) int a[][] = new int [][4];
(d) int []a[] = new int [4][];
(e) int [][]a = new int [4][4];

14)
Which of these array declaration statements are not legal?
Select the two correct answers.
(a) int[] i[] = { { 1, 2 }, { 1 }, {}, { 1, 2, 3 } };
(b) int i[] = new int[2] {1, 2};
(c) int i[][] = new int[][] { {1, 2, 3}, {4, 5, 6} };
(d) int i[][] = { { 1, 2 }, new int[2] };

15.**Using the fragments below, complete the following code so it compiles.**
Note, you may not have to fill all of the slots.
Code:

```
class AgedP {
 _______ _______ _______ _______ _______
 public AgedP(int x) {
 _______ _______ _______ _______ _______
 }
}
public class Kinder extends AgedP {
 _______ _______ _______ _______ _______ _______
 public Kinder(int x) {
 _______ _______ _______ _______ _______();
 }
}
```

Fragments: Use the following fragments zero or more times:

AgedP	super	this
(	)	{
}	;	

```
16. class  Z{
          final static int i;
          static    {
                    i = 0;
          }
          static    {
                    i = 0;
          }
}
```

```
17. . class J        {
          public static void main(String[] args)        {
                    int i = 0;
                    int j = i++  +  test(i)  +  i;
                    System.out.println(i);
                    System.out.println(j);
          }
          static int test(int i){
                    return i++;
          }
}
```

```
18.class F
      {   public static void main(String[] args)
            {       int i = 0;
          int j = i++;
                    i = ++i  +  i--  + -- i  + --j;
                    System.out.println(i);
                    System.out.println(j);
            }       }
```

```
19class G
          {
            public static void main(String args)
          {
                    int i = 100;
                    int  j = i--  + i++  +  i  +  i;
                    System.out.println(i);
                    System.out.println(j);
          }    }
```

ANS:

```
20.
class  R
          {
                    public static void main(String[] args)
                    {
                    int i = 10;
                    int j = ++i  +  test1(++i)  +   test2(i);
                    i=i--  +  test1(i)  +   test2(i--)+i++;
                              System.out.println(i--);
                              System.out.println(--j);
                    }
                    static int test1(int i)
                    {         return ++i;
                    }
                    static int test2(int i)
                    {return  test(i++);
                    }
}
```

```
21.
public class H
  {         public static void main(String[] args)
          {         String s1=new String("SDJ INFOSOFT");
                    System.out.println(s1.substring(2,8));
                    System.out.println(s1.length());
          }
  }
```

ANS:

```
22..public class I
  {
          public static void main(String[] args)
          {
```

```
        StringBuffer s1=new StringBuffer(" hello world ");
                System.out.println(s1.capacity());
                System.out.println(s1.length());
                System.out.println(s1.trimToSize());
                System.out.println(s1.length());
        }
  }
```

ANS:

```
23 public class I  {
public static void main(String[] args)
{
String s1 = "SDJ";
String s2 = "SDJ";
String s3 = new String(s1);
String s4 = new String(s2);
System.out.println("------");
System.out.println(s1 == s2);
System.out.println(s3 == s4);
System.out.println(s1 == s4);
System.out.println(s3 == s4);
System.out.println("--------------");
System.out.println(s1.equals(s2));
System.out.println(s1.equals(s3));
}    }
```

ANS:

```
24. public class T {
public static void main(String[] args)
{
String s1 = "ja";
String s2 = "va";
String s3 = s1.concat(s2);
String s4 = "java";
stem.out.println(s3 == s4);
}     }
25
public class V {
public static void main(String[] args)
{
String s1 = "null";
System.out.println(s1.length());
}   }
```

```
26 public class E {
public static void main(String[] args)
{
String s1 = "abc";
String s2 = "Abc";
String s3 = "xyz";
System.out.println(s1.compareTo(s2));
System.out.println(s1.compareTo(s3));
System.out.println(s1.compareToIgnoreCase(s2));
}      }
```

ANS:

```
27.
public class Manager{
        public  static void main(String args[]){
                String s1 = "---abc 123-hello---";
                s1.replace("-", " ");
                s1.trim();
                System.out.println(s1.length());
        }
  }
```

```
28.
publicclass Manager1 {
        publicstaticvoid main(String[] args) {
                inti = 9;
                i = i / (i-i);
                System.out.println(i);
        }
}
28a) publicclass Manager1  {
        staticvoid test(){
        thrownew Error("error occured");
        }
publicstaticvoid main(String[] args){
try{
        test();
 }
catch(Error e){
         System.out.println("caught");
 }
 }}
29. publicclass Manager1 {
        staticvoid test(){
        thrownewRuntimeException("error occured");
        }
publicstaticvoid main(String[] args){
try{      test();
 }
catch(Error e){
         System.out.println("caught");
 } }}
30. class SdjManager1
      {  public static void main(String []args)
         {           System.out.println(test());
         }
         private  static int test()
         {
       try
        {int   i = 1000 / (1000-1000);
         return 20;
         }
      catch(Throwable t)
         {  return 0;
         }
}
```

```
31. class SdjManager1
      {  public static void main(String []args)
         {  System.out.println(test());
         }
         private  static int test()
         {
       try
        {int   i = 1000 / (1000-1000);
         return 20;
         }
      catch(Throwable t)
```

```
            {  return 0;
            }
          finally
            {return 1000;
            }
            return  2000;
}
```

32.

```
public class SdjManager1
        {   static void test()
            {
               throw new Error(“error occured”);
            }
            public  static void main(String []args)
            {
                try
                {      test();        }
                catch(Error e)
                {       System.out.println(“caught”);    }
            }       }
```

33.

```
public class Test implements Runnable {
public static void main(String[] args) {
new Test();
}
public Test() {
Thread t = new Thread(this);
t.start();
t.start();
}
public void run() {
System.out.println("Test");
}
}
```

a) Test Test b) test IllegalThreadStateException

b) StackOverFlow error c) Test

34.

```
class Thread4 extends Thread{
            publicvoid run()
            {           for (int i = 0; i < 500; i++)
                        {System.out.println(i);
                        }
            }}
publicclass Manager4
{           publicstaticvoid main(String[] args)
throwsInterruptedException
            {
                        Thread4 t4 = newThread4();
                        t4.join();
                        for (int i = 500; i < 1000; i++)
                        {
                                    System.out.println(i);
                        }
                        System.out.println("done");
                                    }}
```

35.

```
class MyThread extends Thread{
public void run(){
System.out.println("Done");
}
public void demo(){
System.out.println("Demo");
}
public static void main(String args[]){
MyThread th=new MyThread();
th.run();
th.stop();
th.demo();
}
}
```

What will happen if you try to compile and run above code:

1. It will throw an exception at th.run() because run() was called before calling start().
2. It will throw an exception at th.demo() because Thread variable th was already stopped calling stop().
3. It will output "Done" following "Demo"
4. Neither of the above.

Ans:

36.

.) What is the output?

```
  FileReader fr  = new FileReader(file);
  char []b = new char[(int)file.length()];
  fr.read(b);
```

a) Reads contents of file into char array.

b) Reads contents of file as bytes into char array.

c) Reads contents of file as Byte object into char array.

d) None of the above.

Ans.

37.

```
           public class  F extends Thread
 {          public static void main(String[] args)
            {           System.out.println("main start");
                        F a1 = new F();
                        a1.start();
                        a1.setPriority(NORM_PRIORITY);
                        System.out.println(a1.getPriority());
                        System.out.println("main end");
            }
            public void start()
            {           System.out.println("start");
            }
                        public void run()
                        {           System.out.println("run");
                        }
            }
```

“Talk to Yourself Once in a Day Otherwise You May Miss Meeting an Excellent Person in this World”

*****swami Vivekananda**

Paper Solution of PF Test 2016 Batch-2

1. CTE
2. CTE can conver int to byte
3. null
4. NO
5. Constructor
Method
6. 2
1
www.javapadho.com
7. CTE
8. Hi
Hello
Hi
9 a b e
10 b e
11 b d
12 a c
13 c
14 b
15 Aged(){} this() super()
16. CTE final value not in
17 1
2
18 3
-1
19
100
399
20 45
35
21
J INFO
12
22 CTE if comment trimTo size line
29
13
13
23
------
true
false
false
false
----
true
true

24 false
25 4
26
32
-23
0
27 19
28 ArithmeticException
28a) caught
29) error occured
30) 0
31) CTE unrechable code
32) caught
33) (b) test IllegalThreadStateException not call start method two time with same object.

34) 500 to 1000 done
run method not call
35) done demo
36) a
37) main start
start
5
main end