PART-A

**B**

**B**

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
FIRST SEMESTER, 2017-2018
COMPREHENSIVE EXAMINATION
PART-B (CLOSED BOOK)
CS F213 OBJECT ORIENTED PROGRAMMING

DATE: 09-12-2017                    MAX DURATION: 60 MIN                    MAX MARKS: 20

## INSTRUCTIONS

(a)  Only one option is correct. Each correct answer carries 1 mark. Each wrong answer carries (-ve) 0.5 marks.
(b)  Overwriting is STRICTLY NOT ALLOWED question will not be corrected.
(c)  Answers marked with pencils will not be corrected.
(d)  Assume that all the required packages and/or classes are imported in the code.
(e)  Ignore errors such as uppercase lowercase errors, semicolon missing, inverted commas missing etc.

### PREDICT THE OUTPUT OF THE FOLLOWING QUESTIONS

**Q.1**
```
   class A implements Runnable{
          public void run(){
                  System.out.println("run-A");
          }
   }
1. public class Test {
2.      public static void main(String argv[]){
3.             A a = new A();
4.             Thread t = new Thread(a);
5.             System.out.println(t.isAlive());
6.             t.start();
7.             System.out.println(t.isAlive());
8.      }}//end of class Test
```

**Q.1 OPTIONS:**
[A] false run-A true
[B] false run-A false ✓
[C] true run-A true
[D] Compilation fails due to an error on line 7

**Q.2**
```
   class MySort implements Comparator<Integer> {
          public int compare(Integer x, Integer y) {
                  return y.compareTo(x);
          }
   }
```
And the code fragment below:
```
Integer[] primes = {2, 7, 5, 3};
MySort ms = new MySort(); Arrays.sort(primes, ms);
for (Integer p2 : primes)  System.out.print(p2 + " ");
```

**Q.2 OPTIONS:**
[A] 2 3 5 7 ✓
[B] 2 7 5 3
[C] 7 5 3 2
[D] Compilation fails

**Q.3**
```
public class Test {
   public static void main(String[] args) {
      Integer sum1 = 125;   //ln 1
      int sum2 = 125;        //ln 2
      System.out.print(sum1.equals(sum2)); //ln 3
      System.out.print(sum2.equals(sum1)); //ln 4
      System.out.print(sum1==sum2); //ln 5
   }}//end of class Test
```

**Q.3 OPTIONS:**
[A] Compilation error at line 3
[B] Compilation error at line 4 ✓
[C] truetruefalse
[D] truetrueture ✓

**Q.4**
```
public class Test {
   public static void main(String[] args) {
      Integer[][] arr1 = { { 1, 2 }, { 3, 4 } };
      Number[] arr2 = arr1[0]; int x = 1;
      System.out.print(arr1[0][0] == 1); ✓
      System.out.print(arr2[0] == x);
      System.out.print(x instanceof Integer); ✓
      System.out.print(arr1[1][0] > (Integer) arr2[0]);
   }}//end of class Test
```

**Q.4 OPTIONS:**
[A] All statements will compile
[B] Only one will compile and will print true
[C] Only two statements will compile and both will print ture ✓
[D] Only three statements will compile, only two of them will print true

**Q.5**
```
1. public class Test {
2.     enum Month { JAN, FEB, MAR };
3.     public static void main(String[] args) {
4.         Month m1 = Month.JAN;
5.         Month m2 = Month.JAN;
6.         Month m3 = Month.FEB;
7.         System.out.println(m1 == m2);
8.         System.out.println(m1.equals(m2));
9.         System.out.println(m1 == m3);
10.        System.out.println(m1.equals(m3));
11.    }
12. }
```

**Q.5 OPTIONS:**
[A] true true true false
[B] true true false false ✓
[C] false false true true
[D] Compilation fails with an error at line 10

**Q.6**
```
public class Test {
    public static void main(String[] args){
        String value = "abc";
        changeValue(value);
        System.out.println(value);
    }
    public static void changeValue(String a){ a = "xyz"; }
}//end of class Test
```

**Q.6 OPTIONS:**
[A] abc ✓
[B] xyz
[C] compilation fails
[D] compilation clean, No output

**Q.7**
```
public class Test 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();
    }}//end of class Test
```

**Q.7 OPTIONS:**
[A] Compilation error, can't invoke start() twice
[B] runrun ✓
[C] IllegalThreadStateException will be thrown because of the second invoke to start()
[D] run

**Q.8 Which is true?**
[A] "X extends Y" is correct if and only if X is a class and Y is an interface
[B] "X extends Y" is correct if and only if X is an interface and Y is a class
[C] "X extends Y" is correct if X and Y are either both classes or both interfaces ✓
[D] "X extends Y" is correct for all combination of X and Y being classes and/or interfaces

**Q.9**
```
public class Test {
    static { System.out.println("static"); }
    { System.out.println("block"); }
    public A(){ System.out.println("A"); }
    public static void main(String[] args){
        A a = new A();
    }}//end of class Test
```

**Q.9 OPTIONS:**
[A] A block static
[B] static block A
[C] static A ✓
[D] A

**Q.10 What three lines will the following code print out when compiled and run?**
```
public class Spindle{
    public static void main(String[] args){
        int[] a = new int[] {1,2,3,4}; int b = 5; int c =6;
        Fold.mutilate(a,b,c);
        System.out.println(a[0]);    System.out.println(b);        System.out.println(c);
}}//end of class Spindle
class Fold {
    static void mutilate( int[] a, int b , int c) { a[0] = 7; b = 8; c = 9; }
}
```

**Q.10 OPTIONS:**
[A] 7,8,9         [B] 7,5,6 ✓         [C] 1,5,6         [D] Error

```
1. public class A {
2.     int add(int i, int j){
3.         return i+j;
4.     }
5. }
```

```
6.  public class B extends A{
7.      public static void main(String argv[]){
8.          short s = 9;
9.          System.out.println(add(s,6));
10.     }
11. }
```

**Q.11 OPTIONS:**
[A]Compile fail: error on line no 2 [B]Compile fail: error on line no 9
[C]Compile fail: error on line no 8 [D]15 ✓

**Q.12**
```
1. public class Test {
2.     public static void main(String... args) {
3.         Integer i = 34;
4.         String str = (i<21)?"jan":(i<56)?"feb":"march";
5.         System.out.println(str);
6.     }
7. }
```

**Q.12 OPTIONS:**
[A] feb ✓
[B] jan
[C] march
[D] Compilation fails
at line no. 4

**Q.13**
```
1. public class Test {
2.     int i=8;
3.     int j=9;
4.     public static void main(String[] args){
5.         add();
6.     }
7.     public static void add(){
8.         int k = i+j;
9.         System.out.println(k);
10.     }
11. }
```

**Q.13 OPTIONS:**
[A] 17 ✓
[B] 0
[C] Compilation fails with an
error at line 5
[D] Compilation fails with an
error at line 8

**Q.14**
```
public class MyScan {
    public static void main(String[] args) {
        String in = "1 a 10 . 100 1000";
        Scanner s = new Scanner(in);
        int accum = 0;
        for (int x = 0; x < 4; x++)
            accum += s.nextInt();
        S.O.P(accum);
    }
}
```

**Q.14 OPTIONS:**
[A] 11
[B] 111
[C] 1111
[D] Runtime exception

**Q.15**
```
public class Test {
    public static void main(String[] args) {
        int x =5;   x *= 3 + 7;
        System.out.println(x);
    }
}
```

**Q.15 OPTIONS:**
[A] 22    [B] 50 ✓  [C] 10
[D] compilation fail with
an error at line 4

**Q.16**
```
1. public class Test {
2.     public static void main(String[] args) {
3.         int [] index = new int[5];
4.         S.O.P(index instanceof Object);
5.     }
6. }
```

**Q.16 OPTIONS:**
[A] true ✓
[B] false
[C] Compilation fails
with an error at line 3
[D] Compilation fails
with an error at line 4

**Q.17 What will you do to the below code to ensure the integrity of data?**

```java
public class SyncTest{
    public static void main (String [] args)  {        class Foo {
        Thread t = new Thread() {                          private int data = 23;
            Foo f = new Foo();                             public void increase(int amt) {
            public void run() { f.increase(20); }              int x = data;
        };                                                     data = x + amt;
        t.start();                                         }
    }                                                  }
}
```

**Q.17 OPTIONS:**

[A]  Synchronize the run method
[B]  Wrap a synchronize(this) around the call to f.increase()
[C]  The existing code will cause a runtime exception
[D]  Synchronize the increase() method ✎

**Q.18**
```java
public class A {
    private void printName(){                  public class B extends A {
        System.out.println("Value-A");             public void printName(){
    }                                                  System.out.println("Name-B");
}//end of class A                                  }
                                               }//end of class B

public class Test{
    public static void main (String[] args) {
        A b = new B();
        b.printName();
    }
}//end of class Test
```

**Q.18 OPTIONS:**
[A] Value-A
[B] Name-B ✎
[C] Value-A Name-B
[D] compilation fail

**Q.19**
```java
public class Bees {
    public static void main(String[] args) {
        try { new Bees().go(); } catch (Exception e) { S.O.P("in main"); }
    }
    synchronized void go() throws InterruptedException {
        Thread t1 = new Thread();
        t1.start();
        System.out.print("1 ");
        t1.wait(5000);
        System.out.print("2 ");
    }
}
```

**Q.19 OPTIONS:**
[A] 1 then 2 after 5 seconds
[B] 1 in main
[C] 1 2 in main
[D] 1 then t1 waits for its notification.

**Q.20**
```java
public class Test {
    public static void main(String[] args) {
        Integer i = 34;        int l = 34;
        if(i.equals(l)) {System.out.println(true);}
        else { System.out.println(false); }
    }
}
```

**Q.20 OPTIONS:**
[A] true ✎      [B] false
[C] Compile error
[D] None of the above