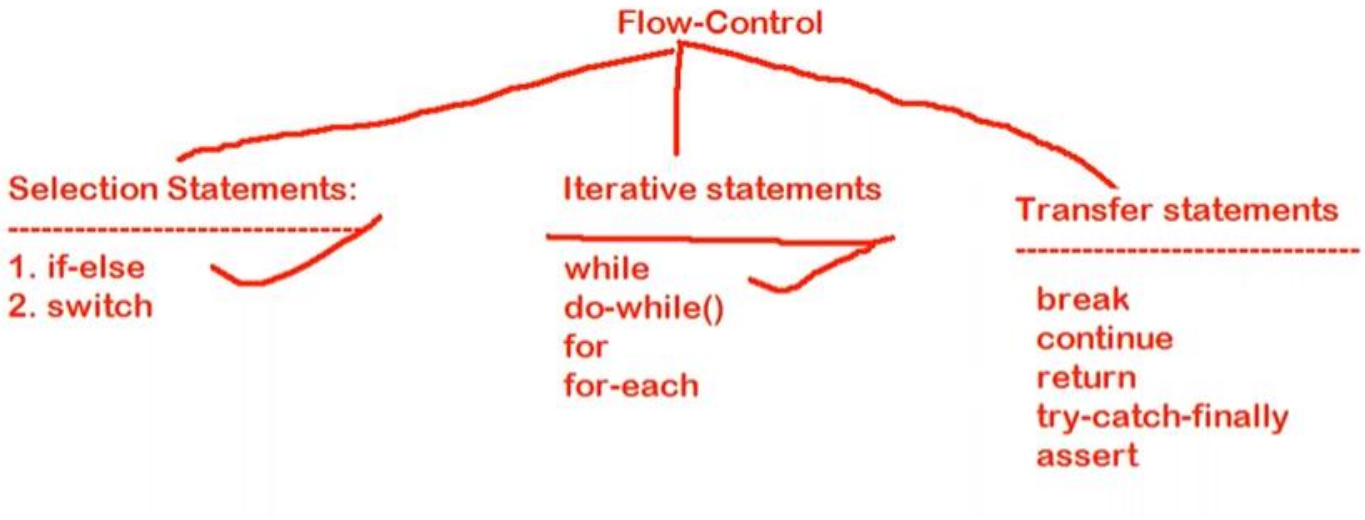
FLOW-CONTROL



```
class Test
{
    public static void main(String[] args)
    {
        int x=1;
        if(x)
        {
            System.out.println("Hello");
        }
        else
        {
            System.out.println("Hi");
        }
    }
}
```

Incomparable type

```
class Test
{
    public static void main(String[] args)
    {
        int x =10;
        int y =20;
        switch(x)
        {
            case 10:
                System.out.println(10);
            case y:
                System.out.println(20);
        }
    }
}
```

```
Test.java:11: error: constant expression required
            case y:
                 ^
```

```
class Test
{
    public static void main(String[] args)
    {
        byte b=10;
        switch(b)
        {
            case 10:
                System.out.println(10);
                break;
            case 100:
                System.out.println(100);
                break;
            case 1000:
                System.out.println(1000);
        }
    }
}
```

```
class Test
{
    public static void main(String[] args)
    {
        int x=1;
        switch(x)
        {
            default:
                System.out.println("def");
            case 0:
                System.out.println(0);
                 break;
            case 1:
                System.out.println(1);
            case 2:
                System.out.println(2);

        }
    }
}
```

Q U E S T I O N

```
public class Test
{
    public static void main(String[] args)
    {
        String stuff="X";
        String res=null;
        if(stuff.equals("X"))
        {
            res="A";
        }
        else if(stuff.equals("Y"))
        {
            res="B";
        }
        else
        {
            res="C";
        }
    }
}
```

Which of the following code can replace nested if-else?
A) res=stuff.equals("X") ? "A" : stuff.equals("Y") ? "B" : "C";
B) res=stuff.equals("X") ? stuff.equals("Y") ? "A" : "B" : "C";
C) res=stuff.equals("X") ? "A" else stuff.equals("Y") ? "B" : "C";
D) res=stuff.equals("X") ? res="A" : stuff.equals("Y") ? "B" : "C";

```
public class Test
{
    public static void main(String[] args)
    {
        //line-1
        switch(x)
        {
            case 10:
                System.out.println("Ten");
                break;
            case 20:
                System.out.println("Twenty");
                break;
        }
    }
}
```

Which 3 code fragments can be independently inserted at line-1 to print Ten
A) byte x =10;
B) short x =10;
C) String x ="10";
D) long x =10;
E) double x =10;
F) Integer x = new Integer(10);

```java
public class Test
{
    public static void main(String[] args)
    {
        boolean b = true;//line-1
        switch(b)
        {
            case true://line-2
                System.out.print("True");
                break;
            default:
                System.out.print("default");
        }
        System.out.println("Done");
    }
}
```

Which of the following changes are required to print TrueDone?

A) Replace line-1 with String b="true";
Replace line-2 with case "true";

B) Replace line-1 with boolean b=1;
Replace line-2 with case 1;

C) remove break statement

D) remove the default section

```java
public class Test
{
    public static void main(String[] args)
    {
        String color="Green";
        switch(color)
        {
            case "Red":
                System.out.println("Red");
            case "Blue":
                System.out.println("Blue");
                break;
            case "Green":
                System.out.println("Green");
                break;
            default:
                System.out.println("Default");
        }
    }
}
```

What is the output?

A) Red
Blue

B) Green
Default

C) Default

D) Green

Q. Which of the following is true about switch statement?

A) It should contain the default section

B) The break statement, at the end of each case block is mandatory

C) Its case lable literals can be changed at runtime

D) Its expression must evaluate a single value ✓

WHILE STATEMENT

```
class Test
{
   public static void main(String[] args)
   {
      while(1)
      {
         System.out.println("Hello World!");
      }
   }
}
```

```
D:\durgaclasses>javac Test.java
Test.java:5: error: incompatible types: int cannot be converted to boolean
                while(1)
                      ^
1 error
```

For Loop

```
class Test
{
   public static void main(String[] args)
   {
      int i=0;
      for(System.out.println("Hello Boss You are Sleeping");i<3;System.out.println("No Boss U only
      sleeping"))
      {
         i++;
      }
   }
}
```

For-each loop

```
for-each loop:
--------------
Enhanced for loop
1.5 version
Specially designed loop to retrieve elements fo arrays and collections
```

Some requirements:

1.. Process all the elements of the array in the order of entry
2.. Process all the elements of the array in the reverse order of entry
3.. Process alternative elements of the array in the order of entry

```
for-loop:1,2,3
for-each:1
```

Break:

break:
---
1. inside switch to stop fall through
2. inside loops to break loop execution based on some condition
3. we can use inside labeled blocksto break block execution

```
1  class Test
2  {
3      public static void main(String[] args)
4      {
5          int x=10;
6          System.out.println("Hello");
7          l1:
8          {
9              System.out.println("Begin");
0              if(x==10)
1                  break l1;
2              System.out.println("End");
3          }
4          System.out.println("Hi");
5      }
6  }
7
```

Labeled block example

Continue

1> Only inside loop

```
public class Test
{
   public static void main(String[] args)
   {
      int i =5;
      while(isAvailable(i))
      {
         System.out.print(i);//Line-1
         //Line-2
      }
   }
   public static boolean isAvailable(int i)
   {
      return i-- > 0 ? true : false;// Line-3
   }
}
```

Which modification enables the code to print 54321

A) Replace Line-1 with System.out.print(--i);
B) At Line-2, insert i--; Replace Line-3 with return (i> 0) ? true : false;
C) Replace Line-1 with --i; and , at Line-2 insert System.out.print(i);
D) Replace Line-3 with return (i> 0) ? true : false;

Which modification enables the code to print 54321

A) Replace Line-1 with System.out.print(--i);
B) At Line-2, insert i--;
C) Replace Line-1 with --i; and , at Line-2 insert System.out.print(i);
D) Replace Line-3 with return (i> 0) ? true : false;

```
public class Test
{
   public static void main(String[] args)
   {
      int[] x= {1,2,3,4};
      int i =0;
      do
      {
         System.out.print(x[i]+" ");//123
         i++;
      }
      while (i<3);
   }
}
```

```
1  public class Test
2  {
3     public static void main(String[] args)
4     {
5        int x =5;
6        do
7        {
8           System.out.print(x-- +" ");
9        }
0        while (x==0);
1     }
2  }
```

What is the result?

A) 5 4 3 2 1 0
B) 5 4 3 2 1
C) 4 2 1
D) 5
E) Nothing is printed

```java
public class Test
{
    public static void main(String[] args)
    {
        int[][] x = new int[2][4];
        x[0]= new int[]{2,4,6,8};
        x[1]= new int[]{2,4};
        for(int[] x1: x)
        {
            for(int x2 :x1)
            {
                System.out.print(x2+" ");
            }
            System.out.println();
        }
    }
}
```

```
2468
24
```

```java
public class Student
{
    String name;
    public Student(String name)
    {
        this.name=name;
    }
}
public class Test
{
    public static void main(String[] args)
    {
        Student[] s = new Student[3];
        s[1]= new Student("Durga");
        s[2]= new Student("Ravi");
        for(Student s1: s)
        {
            System.out.println(s1.name);
        }
    }
}
```

What is the output?

A) Durga
Ravi

B) Durga
Ravi
null

C) Compilation Fails

D) ArrayIndexOutOfBoundsException

E) NullPointerException

```
public class Test
{
   public static void main(String[] args)
   {
      int[]  data={10,20,30,40,50,30};
      int k= 30;
      int count=0;
      for(int x : data)
      {
         if( x!= k)
         {
            continue;
            count++;
         }
      }
      System.out.println(count);
   }
}
```

UNREACHABLE

What is the result?

A) 0
B) 1
C) 2
D) Compilation Fails

```
public class Test
{
   public static void main(String[] args)
   {
      int wd = 0;
      String[] days={"sun","mon","wed","sat"};
      for(String s : days)
      {
         switch(s)
         {
            case "sat":
            case "sun":
               wd -= 1;
               break;
            case "mon":
               wd++;
            case "wed":
               wd += 2;
         }
      }
      System.out.println(wd);
   }
```

What is the output?

A) 3
B) 4
C) -1
D) Compilation Fails

```
public class Test
{
    public static void main(String[] args)
    {
        int[] x= {1,2,3,4,5}
        for(yyy)
        {
            System.out.print(a[i]);
        }
    }
}
```

Which option can replace yyy to enable the code to print 135?

A) int i=0;i<=4;i++
B) int i=0;i<5;i+=2
C) int i=1;i<=5;i++
D) int i=1;i<5;i+=2