Flow of Control

Type A: Very Short Answer Questions

1	**What is the significance of a null statement?**
Ans.	A **null statement** is useful in those instances where the syntax of the language requires the presence of a statement but where the logic of the program does not.
2	**What are the three constructs that govern statement flow?**
Ans.	1. Sequence 2. Selection 3. Iteration
3	**What is a selection statement? Which selection statements does java provide? Write one advantage and one disadvantage of using ? : in place of if.**
Ans.	The **selection statements** allow to choose the set-of-instructions for execution depending upon an expression's truth value. Java provides two types of selection statements **–if** and **switch.** **Advantage –**?: offers more concise, clean and compact code. **Disadvantage –**It is less obvious as compared to if.
4	**Can a conditional operator replace any if statement always? Suggest a way to override the default dangling-else matching.**
Ans.	No, one method of overriding the default dangling-else matching is to place the occurring unmatched if in a compound statement.
5	**In a nested-if, how does the default matching of dangling-else take place?**
Ans.	In nested if statement, a dangling else statement goes with the preceding unmatched if statement.
6	**One out of several different alternatives can be selected with the help of which statement?**
Ans.	Switch statement
7	**What is significance of a break statement in a switch statement?**
Ans.	The break statements are necessary because without them, statements in switch blocks fall through: All statements after the matching case label are executed in sequence, regardless of the expression of subsequent case labels, until a break statement is encountered.
8	**Write one limitation and one advantage of a switch statement.**
Ans.	**Advantage:** ✓ The switch statement is a control statement that handles multiple selections and enumerations by passing control to one of the case statements within its body. **Limitation:** ✓ Logical operators cannot be used with switch statement. case k>=20; ✓ Switch case variables can have only int and char data type.
9	**What is the effect of absence of break in a switch statement?**
Ans.	without the break statement, statements in switch blocks fall through
10	**The case lables in a switch can have identical values. (T/F)?**
Ans.	False
11	**What is the significance of default clause in a switch statement?**
Ans.	Default clause is used for the else situation in a switch statement
12	**What are iteration statements? Name the iteration statements provide by java.**
Ans.	The **iteration statements** allow a set of instructions to be performed repeatedly until a certain condition is fulfilled. Java provides three kind of looping statements: for loop, while loop, do-while loop.
13	**Which elements are needed to control a loop?**
Ans.	1. Initialization Expression(s). 2. Test Expression. 3. Update Expression(s). 4. The Body-of-the-Loop.
14	**What is meant by an entry-controlled loop? Which java loops are entry-controlled?**

Ans.	**Entry-controlled loops –** Here loop condition is tested before entering in to the loop. for and while loops are entry-controlled loop.
15	**What is meant by an exit-controlled loop? Which java loops are exit-controlled?**
Ans.	**Exit-controlled loops –**Here loop body is executed first and then continuation condition is evaluated. Do-while loop is exit-controlled loop.
16	**The update expression in a for loop can decrement the loop variable. T/F?**
Ans.	True
17	**The initialization expression of a for loop must be followed by a semicolon. T/F?**
Ans.	True
18	**The update expression of a for loop must be followed by a semicolon. T/F?**
Ans.	False
19	**Write for loop that displays the numbers from 51 to 60.**
Ans.	for(int i=51;i<60;i++)
20	**Write for loop that displays the numbers from 10 to 1 i.e., 10, 9, 8…….. 3, 2, 1**
Ans.	for(i=10;i>=1;i--)
21	**Which expressions are optional in a for loop? When are empty loops useful?**
Ans.	In a for loop, initialization expression, test expression and update expression are optional. An empty loop being used for time delay.
22	**What is meant by a variable's scope?**
Ans.	A **variable scope** is the part of program within which you can access the variable. A variable scope is the block of code where the variable has been declared.
23	**What is the difference between a while and do-while loop?**

Ans.

while	**do-while**
✓ While is an entry-controlled loop.	✓ do-while is a exit-controlled loop.
✓ In while loop the test expression is evaluated at the end of the loop i.e., after executing the loop body.	✓ In do-while loop the test expression is evaluated at the beginning of the loop i.e., before executing the loop body.

✓ Syntax:
while(test-expression)
{
// statement
}
	✓ Syntax:
	do
	{
	// statement
	} while(test-expression);

24	**How is break statement different from a labeled break statement?**
Ans.	break cause the flow of control to jump to the statement immediately following the current statement. Break label cause flow of control to break out of the containing statement which must be labeled label.
25	**When are labeled loops useful?**
Ans.	Labeled loops are useful with break and continue.
26	**A code displaying a menu must be executed at least once. Which loop is most suitable for it?**
Ans.	do-while loop.

Type B: Short Answer Questions

1	**What is the problem of dangling-else? When does it arise? What is the default dangling-else matching and how can it be overridden?**

Ans.	The nested if-else statement introduces a source of potential ambiguity referred to as dangling-else problem.
	This problem arises when in a nested if statement, number of ifs is more than the number of else clause.
	if (expr1)
	if((expr2)
	statement –1;
	else
	statement –2;
	You want the else to go with the other if. By default, it will go with the inner if.

	One method of overriding the default dangling-else matching is to place the occurring unmatched if in a compound statement.
2	**Compare an if and a ? : operator.**
Ans	1. Compared to if-else sequence, ?: offers more concise, clean and compact code, but it is less obvious as compared to if. 2. Another difference is that the conditional operator ?: produces an expression, and hence a single value can be assigned or incorporated into a larger expression, whereas, if is more flexible. The if statement can have multiple statements, multiple assignments and expressions in its body.
3	**Given the following code fragment:** **if (a == 0)** **System.out.println ("Zero");** **if (a == 1)** **System.out.println ("One");** **if (a == 2)** **System.out.println ("Two");** **if (a == 3)** **System.out.println ("Three");** **Write an alternative code (using if) that saves on number of comparisons.**

Ans.	`if(a==0)`
	`{`
	`System.out.println("Zero");`
	`}`
	`else if(a==1)`
	`{`
	`System.out.println("One");`
	`}`
	`else if(a==2)`
	`{`
	`System.out.println("Two");`
	`}`
	`else if (a == 3)`
	`{`
	`System.out.println("Three");`
	`}`

4	**Rewrite the following fragment using switch:** **if (ch == 'E')** **eastern++;** **if (ch == 'W')** **western++;** **if (ch == 'N')** **northern++;** **if (ch == 'S')** **southern++;** **else** **unknown++;**

Ans.	`switch(ch)`
	`{`
	`case'E': eastern++;`
	`break;`
	`case'W': western++;`
	`break;`
	`case'N': northern++;`
	`break;`
	`case'S': southern++;`
	`break;`

	default:
	unknown++;
	}

5	**Write the syntax and purpose of a switch statement.**

Ans.	**Syntax:**
	switch (expression)
	{
	case constant1: Statement sequence 1;
	break;
	case constant2: Statement sequence 2;
	break;
	case constant3: Statement sequence 3;
	break;
	:
	case constant n-1: Statement sequence n-1;
	break;
	[default: statement sequence n];
	}
	Purpose –This selection statement successively tests the value of an expression against a list of integer or character constants.

6	**When does an if statement prove more advantageous over a switch statement?**
Ans.	Switch can only test for equality whereas if can evaluate a relation or logical expression i.e, multiple conditions.
7	**When does a switch statement prove more advantageous over an if statement?**
Ans.	The switch statement is more efficient than if in a situation that supports the nature of switch operation.
8	**Why is it suggested to put a break statement after the last case statement in switch even though it is not needed syntactically?**
Ans.	It is not necessary to put a break after the last statement in a switch, since control will leave the statement anyway, yet it should be done to avoid forgetting the break when you add another case statement at the end of the switch.
9	**Rewrite the code given in question 3 using switch.**

Ans.	switch(a)
	{
	case 0: System.out.println(“Zero”);
	break;
	case 1: System.out.println(“One”);
	break;
	case 2: System.out.println(“Two”);
	break;
	case 3: System.out.println(“Three”);
	break;
	}

10	**Rewrite the following set of if-else statements in term of switch-case statement:** **(a) char code;** **code = character.readChar();** **if(code == 'A')** **System.out.println("Accountant");** **else if (code == 'C' || code == 'G')** **System.out.println("Grade IV");** **else if (code == 'F')** **System.out.println("Financial Advisor");** **(b) int inputnum, calcval;** **if(inputnum == 5)** **{** **calcval = inputnum * 25 -20;**

	System.out.println(inputnum + calcval); **}** **else if (inputnum == 10)** **{** **calcval=inputnum *25 -20;** **System.out.println(calcval - inputnum);** **}**

Ans.	**(a)** char code;
	code = character.readChar();
	switch(code)
	{
	case 'A': System.out.println("Accountant");
	break;
	case 'C': System.out.println("Grade IV");
	break;
	case 'G': System.out.println("Grade IV");
	break;
	case 'F': System.out.println("Financial Advisor");
	break;
	}
	(b) int inputnum, calcval;
	switch(inputnum)
	{
	case 5: calcval = inputnum * 25 -20;
	System.out.println(inputnum + calcval);
	break;
	case 10: calcval=inputnum *25 -20;
	System.out.println(calcval - inputnum);
	break;
	}

11	**How many times are the following loops executed?** **(a) X=5; Y=50;** **while(X <= Y)** **{** **X = Y/X;** **…….** **}** **(b) int m=10,n=7;** **while (m % n >=0)** **{** **. .. . .** **m =m + 1;** **n = n + 2;** **……** **}**
Ans.	(a) Infinite loop (b) Infinite loop
12	**Given the following code fragment:** **i = 2;** **start:** **System.out.println (i);** **i += 2;** **if (i < 51) goto start;**

	System.out.println ("Thank you"); **Rewrite the above code using a while loop.**

Ans.	i = 2;
	while (i < 51);
	{
	i += 2;
	System.out.println(i);
	}
	System.out.println ("Thank you");

13	**Given the following code fragment:** **i = 100;** **while (i > 0)** **System.out.println (i--)** **System.out.println ("Thank you");** **Rewrite the above code using a do while loop.**

Ans.	i = 100;
	do{
	System.out.println (i);
	i--;
	}while (i > 0)
	System.out.println ("Thank you");

14	**Rewrite following code using while loop** **int sum = 0;** **for (int i = 1 ; i <= 5 ; ++i)** **{** **sum = sum + c;** **//variable c is not declared In book also, so in answer c is replaced with i** **}**

Ans.	int sum = 0,i=1;
	while(i<=5)
	{
	sum = sum + i;
	++i;
	}

15	**Rewrite following while loop into a for loop** **int stripes = 0;** **while (stripes <= 13) {** **if (stripes % 2 == 2)** **{** **System.out.println("colour code Red");** **}** **else {** **System.out.println("colour code Blue");** **}** **System.out.println("New stripe");** **stripes = stripes + 1;** **}**

Ans.	int stripes = 0;
	for(stripes=0; stripes <= 13; stripes++)
	{
	if (stripes % 2 == 2)
	{

```
        System.out.println("colour code Red");
    }
    else
    {
        System.out.println("colour code Blue");
    }
    System.out.println("New stripe");
}
```

16 **Rewrite the following code using either while or do-while loop or both loops.**

```
for(int i=1;i<4;++i)
{
        for (int j=3;j>0;--j)
        {
                System.out.print("###..");
        }
        System.out.println();
}
```

Ans.

```
//while
int i=1;
 while(i<4)
 {
         ++i;
         int j=3;
         while(j>0)
         {
               --j;
                System.out.print("###..");
          }
           System.out.println();
}

//do while
int i=1;
do
{
         ++i;
          int j=3;
          do
          {
              System.out.print("###..");
              --j;
           }while(j>0);
           System.out.println();
       }while(i<4);
```

Output and Error Questions

17 **Find the output of the following code fragments?**

(a)

```
int s = 14;
if (s < 20)
  System.out.print("under");
else
  System.out.print("Over");
  System.out.println("the limit.")
```

(b)

```
int s = 14;
if(s < 20)
```

	System.outprint("under"); **else {** **System.out.print("over");** **System.out.println("the limit");** **}** **(c) int s = 94;** **if(s < 20){** **System.outprint("under");** **}** **else {** **System.out.print("over");** **}** **System.out.println("the limit");**

Ans.	**(a) Compilation Error**
	Correct Code
	`int s = 14;`
	`if (s < 20)`
	`System.out.print("under");`
	`else`
	`System.out.print("Over");`
	`System.out.println("the limit.");`
	Correct Output-> underthe limit
	(b) Compilation Error
	Correct Code
	`int s = 14;`
	`if(s < 20)`
	`System.out.print("under");`
	`else {`
	`System.out.print("over");`
	`System.out.println("the limit");`
	`}`
	Correct Output-> underthe limit
	(c) Output -> Overthe limit

18	**What will be the output of following code fragment when the value of ch is** **(a) 'A' (b) 'C' (c) 'D' (d) 'F'?** **switch(ch) {** **case 'A' : System.out.println("Grade A");** **case 'B' : System.out.println("Grade B");** **case 'C' : System.out.println("Grade C");** **break;** **case 'D' : System.out.println("Grade D");** **default : System.out.println "Grade F";** **}**
Ans.	**(a)** When input is A, the output will be as follows: Grade A Grade B Grade C **(b)** When input is C, the output will be as follows: Grade C **(c)** When input is D, the output will be as follows: Grade D **(d)** When input is F, the output will be as follows: Grade F

19	**Predict the output of the following code fragment:** **int n=0; int j=7; int i=3;** **if(i>2) {** **n=1;** **if(j>4)** **n=2;** **else** **n=3;** **}** **else {** **n=4;** **if(j%2 >= 2)** **n=5;** **else** **n=6;** **}** **System.out.println(n);**
Ans.	2
20	**Predict the output of the following code fragments:** **(a) int i,j,n;** **n=0; i=1;** **do {** **n++; i++;** **} while(i<=5);** **(b) int i=1,j=0,n=0;** **while(i<4) {** **for(j=1;j<=i;j++) {** **n+=1;** **}** **i=i+1;** **}** **System.out.println(n);** **(c) int i=3,n=0;** **while (i<4) {** **n++; i--;** **}** **System.out.println(n);** **(d) int j=1,s=0;** **while(j<10) {** **System.out.println(j+ "+");** **s=s+j;** **j=j+j%3;** **}** **System.out.println("=" +s);**
Ans.	(a) If System.out.print() is inside while loop output is 12345 and if outside of while loop output is 5 (b) 6 (c) -2147483644 (d) 2+ 4+ 5+ 7+

	8+ =27
21	**Find out errors, if any:** **(a) m=1;** **n=0;** **for(;m+n<19;++n)** **System.out.println("Hello \n");** **m=m+10;** **(b) while(ctr !=10); {** **ctr=1;** **sum=sum+a;** **ctr=ctr+1;** **}** **(c) for(a=1, a>10;a=a+1)** **{ .....** **}**

Ans.	(a) int m,n;
	m=1;
	n=0;
	for(;m+n<19;++n)
	System.out.println("Hello \n");
	m=m+10;
	(b) while(ctr !=10) {
	ctr=1;
	sum=sum+a;
	ctr=ctr+1;
	}
	(c) for(a=1; a>10;a=a+1)
	{ .....
	}

22	**Identify the possible error(s) in the following code fragment: Discuss the reason(s) of error(s) and correct the code.** **:** **f=1;** **for(int a=40; (a); a--)** **f*=a;** **:** **s=0;** **for(int a=1;a<40/a++)** **s+=a;**

Ans.	:
	f=1;
	for(int a=40;a>1;a--)
	f*=a;
	:
	s=0;
	for(int a=1;a<40;a++)
	s+=a;

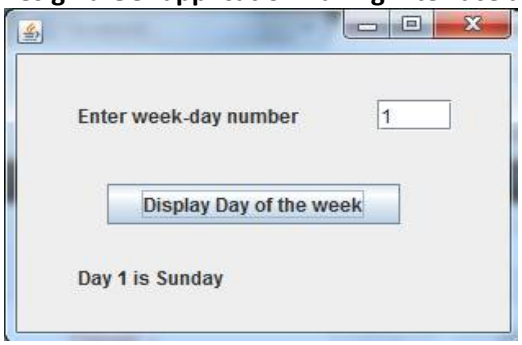
23	**Identify the possible error(s) in the following code fragment. Discuss the reason(s) of error(s) and correct the code.** **while(i>j)** **System.out.println(i*j);** **i++;**
Ans.	The errors in the above code fragment are:

	1. Uninitialized variables i and j are being used in while's test expression. The variables i and j must have a value before it is used in an expression.
	2. The while loop is an infinite loop.
	Correct code:
	`int i=1,j=5;`
	`while(i<j)`
	`{`
	`  System.out.println(i*j);`
	`  i++;`
	`}`

24	**Identify the possible error(s) in the following code fragment. Discuss the reason(s) of error(s) and correct the code.** **while(i>j);** **{** **System.out.println(i*j);** **i++;** **}**

Ans.	The errors in the above code fragment will give compilation is same as Q No. 23. But we have to remove ' ;' after while loop otherwise loop will become infinite loop.
	Correct code:
	`int i=1,j=5;`
	`while(i<j)`
	`{`
	`  System.out.println(i*j);`
	`  i++;`
	`}`

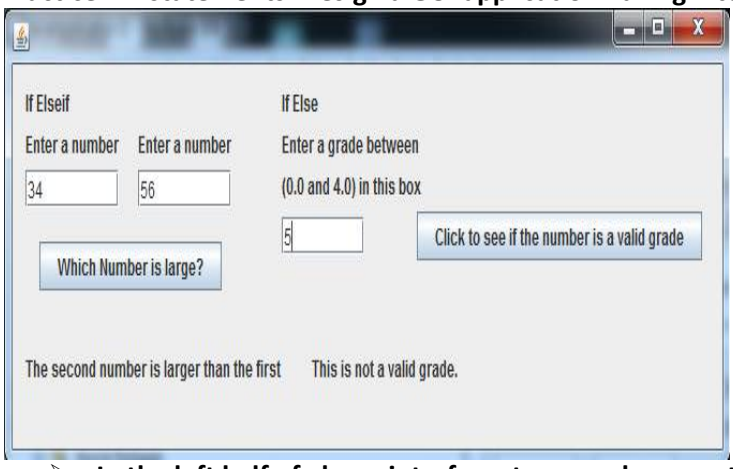
Type c: Practical/Lab Questions

1	**Design a GUI application having interface as shown below:**  **Upon clicking the push button, appropriate weekday should get displayed in the label.**

Ans.	`private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)`
	`{`
	`  int n=Integer.parseInt(jTextField1.getText());`
	`  switch(n)`
	`  {`
	`  case 1:jLabel2.setText("Day 1 is Sunday");`
	`  break;`
	`  case 2: jLabel2.setText("Day 2 is Monday");`
	`  break;`
	`  case 3:jLabel2.setText("Day 3 is Tuesday");`
	`  break;`
	`  case 4: jLabel2.setText("Day 4 is Wednesday");`
	`  break;`
	`  case 5:jLabel2.setText("Day 5 is Thursday");`
	`  break;`
	`  case 6: jLabel2.setText("Day 6 is Frieday");`

```
                    break;
            case 7: jLabel2.setText("Day 7 is Saturday");
                    break;
        }
    }
```

2 **Practice – if statements. Design a GUI application having interface as shown below:**

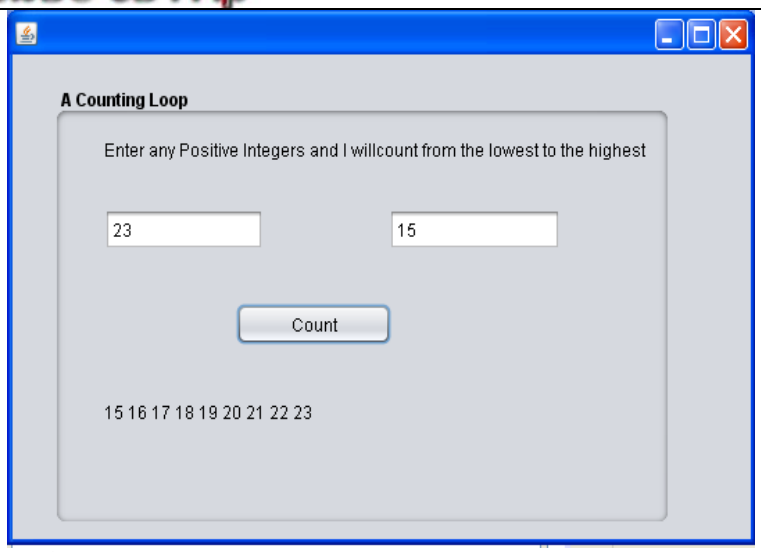


- **In the left half of above interface, two numbers are to be accepted. When the user clicks at button "Which number is larger ?", appropriate message gets displayed in the left label.**
- **In the right half of above interface, a number grade is input. Valid grades are 0-4. Upon clicking the right push button, a message depicting validity of grade is displayed in right label.**

Ans.

```
//larger number
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
        int num1=Integer.parseInt(jTextField1.getText());
        int num2=Integer.parseInt(jTextField2.getText());
        if(num1>num2)
        {
            jLabel4.setText("The first number is larger than the second");
        }
        else
        {
            jLabel4.setText("The second number is larger than the first");
        }
    }
// valid number
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
        float num1=Float.parseFloat(jTextField3.getText());
        if(num1>=0.0 && num1<=4.0)
        {
            jLabel8.setText("This is a valid Grade");
        }
        else
        {
            jLabel8.setText("This is NOT a valid Grade");
        }
   }
```

3 **Practice –Loops. Design a GUI application having interface as shown below:**



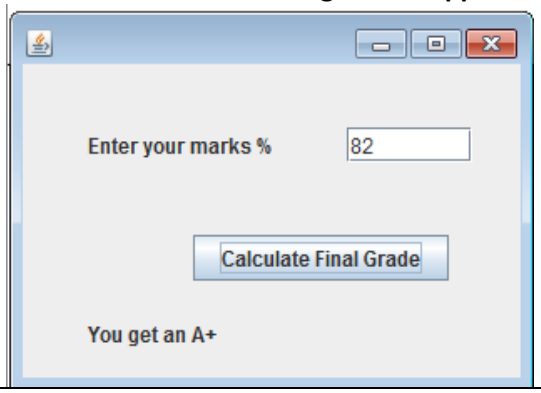
- **The loop for the frame should print numbers from lowest to highest on the basis of given two numbers, upon clicking at "Count" button.**

Ans.

```
// loop count
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
 {
        int num1=Integer.parseInt(jTextField1.getText());
        int num2=Integer.parseInt(jTextField2.getText());
        String msg="";
        if(num1<num2)
        {

            jLabel1.setText(Integer.toString(num1));
            for(int i=num1;i<=num2;i++)
            {
              msg+=Integer.toString(i)+" ";
        }
        jLabel1.setText(msg);
        }
        else
        {
            jLabel1.setText(Integer.toString(num2));
        for(int i=num2;i<=num1;i++)
            {
              msg+=Integer.toString(i)+" ";
        }
        jLabel1.setText(msg);
        }
 }
```

4 **Practice –Select case. Design a GUI application having interface as shown below:**



Marks %	Grade
>=90	A++
80-90	A+
75-80	A
60-75	B
50-60	C
40-50	D
<40	Fail

The percentage marks are to be entered in the text box and upon clicking at the button, corresponding grade (as per following rules) should be displayed in the picture box below command button.

Ans

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
      int num=Integer.parseInt(jTextField1.getText());
      switch(num/10)
      {
          case 10:
              jLabel1.setText("You get an A++");
              break;
          case 9:
              jLabel1.setText("You get an A++");
              break;
          case 8:
              jLabel1.setText("You get an A+");
              break;
          case 7:
              jLabel1.setText("You get an A");
              break;
          case 6:
              jLabel1.setText("You get a B");
              break;
          case 5:
              jLabel1.setText("You get a C");
              break;
          case 4:
              jLabel1.setText("You get a D");
              break;
          default:
              jLabel1.setText("Fail");
      }
    }
```