DELHI PUBLIC SCHOOL NEWTOWN
SESSION: 2021-2022
HALF YEARLY EXAMINATION (ONLINE)

CLASS: IX **FULL MARKS: 40**
SUBJECT: COMPUTER APPLICATIONS (PAPER 2) **TIME: 30 minutes**

Instructions:

- **Attempt all questions.**
- **This paper consists of ten printed pages.**
- **All working including rough work must be clearly shown on the same sheet as therest of the answers.**

Question 1

a) A constant which gives the exact representation of data is called: [1]

i) Variable

ii) Literal

iii) Identifier

iv) Character

Answer: (ii)

b) Name the type of error in each case **[2]**

1) Division by a variable that contains a value of zero.

2) Multiplication operator used when the operation should be division.

i) Runtime, Logical

ii) Runtime, Compile time

iii) Logical, Compile time

iv) Compile time, Runtime

Answer: (i)

c) Predict the output: **[2]**

```
System.out.println(Math.sqrt(Math.min(42.5,42.25)));
```

i) 6.0

ii) 7.0

iii) 6.5

iv) 7.5

Answer: (iii)

Explanation

Math.min(42.5,42.25) gives 42.25. Math.sqrt(42.25) gives output as 6.5.

d) An if-else construct accomplishes 'fall through'. [1]

i) True

ii) False

Answer: (ii)

e) A java statement to import *citrus* class from *Fruit* package where the Fruit package is in the *Eatables* package. [2]

i) Import java.Fruit.Eatables

ii) import java.Eatables.Fruit.citrus;

iii) import Eatables.Fruit.citrus.*;

iv) import java.Eatables.Fruit.citrus.*;

Answer: (ii)

f) Write the equivalent java expression: [1]

$q = 1 / \sqrt{(a + b)} + 3 / c^2$

i) q=1/Math.sqrt(a+b)+3/c

ii) q=1/Math.sqrt (a)+Math.sqrt(b)+3/c*c;

iii) q=1/Math.sqrt(a+b)+3/Math.pow(c)

iv) q=1/Math.sqrt(a+b)+3/Math.pow(c,2);

Answer: (iv)

g) Write the syntax for the functions: **[1]**

To find the absolute value of a number 'y'

i) Math.abs(y)

ii) Math.exp(y)

iii) y=Math.abs( );

iv) Math.aps(y);

Answer: (i)

h) Predict output **[2]**

```
int num=20;
switch(num)
{
case 10: System.out.println("TEN"); break;
case 20: System.out.println("TWENTY"); break;
case 30: System.out.println("THIRTY");
}
```

i TEN
ii) TWENTY
iii) THIRTY
iv) TEN TWENTY

i)

```
int i;
  for(i = 5; i > 10; i++)
  System.out.println(i);
  System.out.println(i * 4);
```

i) 0

ii) 20

iii) 5

iv) 10

Answer: (ii)

j) **[2]**

```
x = 1; y = 1;
  if(n>0)
```

{ x = x + 1;

y = y + 1; }

What will be the value of x and y, if n assumes a value of -3?

i) 2,2

ii) 1,1

iii) 0,0

iv) none of the above

Answer (ii)

k) Predict the output: [2]

```
x = 5; y = 50;
while(x<=y)
{   y = y / x;
    System.out.print(y);  }
```

i) 102

ii) 210

iii) 012

iv) 120

Answer: (i)

l) Predict the value of 'c' [2]

```
int a=6,b=5,c;
c = (a++ % b++) *a + ++a*b++;
```

i) 54

ii) 56

iii) 55

iv) 53

Answer (iii)

m) Predict the number of times the loop runs: [2]

```
class Test
```

```
{   public static void main( )
    {  int i;
       for(i=0;i<5;i++)
       System.out.println(i-i*i);     } }
```

i) 4

ii) 5

iii) 6

iv) 3

Answer: (ii)

n) *If((p>q)&&(q>r)) then* **[1]**

(where p, q and r are three integer numbers)

i) q is the smallest number
ii) q is the greatest number
iii) p is the greatest number
iv) none

Answer (iii)

o) A compound statement can be stated as [1]

i)
```
p=in.nextInt();
q=in.nextInt();
```

ii)
```
m=++a;
n=--b;
```

iii)
```
if(a>b)
{a++;b--;}
```

iv) none

Answer: (iii)

p) What is the differenence between a While and a Do-While loop in java? [1]

i) WHILE loop executes the statements inside of it at least once even if the condition is false.
ii) DO-WHILE loop executes the statements inside of it at least once even if the condition is false.
iii) WHILE loop is fast.
iv) DO-WHILE loop is fast.
Answer: (ii)

Explanation: Both the while and do while loop work at the same speed. A do while loop executes the statements inside of it even when the condition is false. It is the reason why a do while loop is used in menu driven console java programs.

q) Predict the output: [2]

```
for(int j=0;j<5;j++;)
System.out.print(j + “,”);
```

i) 1,2,3,4,
ii) 0,1,2,3,4
iii) Compiler error
iv) None
Answer: (iii)
Explanation:
The semicolon after the INCREMENT/DECREMENT part is not allowed.

r) Predict the output: [2]

```
int score=1;
for(; true; score++)
{  System.out.print(score +",");
   if(score > 3)
   break;
}
```

i) 1,2,3,

ii) 1,2,3

iii) 1,2,3,4,

iv) 1,2,3,4

Answer: (iii)

Explanation: break condition is checked after printing the variable score. So it prints 4 also.

s) To execute a loop 10 times, which of the following statement satisfies: [1]

i) for(i=6;i<=26;i=i+2)
ii) for(i=3;i<=30;i=i+3)
iii) for(i=0;i<10;i=i++)
iv) all of the above

Answer (ii)

t) The ASCII codes of upper case alphabets range from: **[1]**

i) 65 - 90
ii) 60 - 85
iii) 65 - 91
iv) 97 – 122

Answer (i)

u) Which of the following is a correct representation? **[1]**

i) boolean m=true

ii) boolean m='true'

iii) boolean m="true"

iv) none

Answer (i)

v) If int a = 25, b = 5, c = 0; what value is stored in c? When c = a % b; **[1]**

i) 5.0

ii) 5

iii) 0

iv) none

Answer: (iii)

w) Rewrite the following using ternary operator: **[1]**

```
if(a>b)
c=a;
else
c=b;
```

i) c= (b>a)?a:b;
ii) c= (a!=b)?a:b;
iii) c= (a>b)?b:a;
iv) **None**

Answer (iv)

x) *Predict the output:* *[2]*

```
public class Test {
public static void main(String[] args) {
   int count = 1;
   while (count <= 15) {
   System.out.println(count % 2 == 1 ? "***" : "+++++");
   ++count;
      }
   }
}
```

i) 15 times *

ii) 15 times +++++

iii) 8 times * and 7 times +++++**

iv) Both will print only once

Answer: (iii)

Explanation: In the above code, we have declared count = 1. The value of count will be increased till 14 because of the while (count<=15) statement. If the remainder is equal to 1 on dividing the count by 2, it will print (***) else print (+++++). Therefore, for all odd numbers till 15 (1, 3, 5, 7, 9, 11, 13, 15), it will print (***), and for all even numbers till 14 (2, 4, 6, 8, 10, 12, 14) it will print (+++++).

Hence, an asterisk (***) will be printed eight times, and plus (+++++) will be printed seven times.

y) Prime number **is a number that is greater than 1 and divided by 1 or itself only. In other words, prime numbers can't be divided by other numbers than itself or 1. For example 2, 3, 5, 7, 11, 13, 17.... are the prime numbers.**

The following programs checks if the number 'n' is prime or not. Fill the gaps from 1 to 4 to implement the program. **[4]**

```
public class PrimeExample{
 public static void main(String args[]){
  int i,m=0,flag=0;
  int n=3;//it is the number to be checked
  m=__1__;
  if(n==0||n==1){
```

```
  System.out.println(n+" is not prime number");
 }else{
  for(i=2;i<=m;i++){
   if(__2__){
    System.out.println(n+" is not prime number");
    flag=1;
    __3__;
   }
  }
  if(__4__)  { System.out.println(n+" is prime number"); }
 }//end of else
}
}
```

i) n/2, n%i==0, break, flag==0

ii) n/2, n%i==0, break, flag==1

iii) n/2, n/i==0, break, flag==0

iv) n/2, n%i==0, continue, flag==0

Answer (i)

z) Predict output:

```
int persons=45;

int random=45;

switch(45)

{case persons: System.out.print(“CRICKET”);

default: System.out.println(“RUGBY”);

}
```

i) CRICKET

ii) CRICKET RUGBY

iii) RUGBY

iv) Compiler Error

Answer (iv)

Error: case expressions must be constant expressions

So, make the variable final.

final int persons = 45;

//Then, output will be

CRICKET