**1.What will be the output for the following program?**

|  |
| --- |
| public  class Test {  public      static void main(String[] args)      {          for (int i=0;true;i++)              System.out.println("GEEKS");      }  } |

Options:  
1.GEEKS  
2.Compile time error  
3.Run time Exception  
4.GEEKS (Infinitely)

The answer is option (4)

**Explanation:** In the above example, we are using for loop. In for loop if we did not provide any initialization, condition-check and increment/decrement part then it will go to infinite loop if we did not provide any condition in statement.

**2. What will be the output for the following program?**

|  |
| --- |
| class Test {  public      static void main(String[] args)      {          for (int i = 0; i < 3;)              System.out.println("GEEKS");      }  } |

Options:  
1.GEEKS GEEKS GEEKS  
2.Compile time error  
3.GEEKS (Infinitely)  
4.No output

The answer is option (3)

**Explanation:** When we are not taking any statement in increment/decrement section therefore overtime it does not increment/decrement the value of I and the condition always true. That’s why it results into GEEKS (Infinitely).

**3.What will be the output for the following program?**

|  |
| --- |
| class Test {  public      static void main(String[] args)      {          boolean b = true;          if (b = false) {              System.out.println("HELLO");          } else {              System.out.println("BYE");          }      }  } |

Option:  
1.HELLO  
2.BYE  
3.Compile time error: re- initialization  
4.No output

The answer is option (2)

**Explanation :** In the condition of if statement, we assigning are false to b which return a boolean value which is false. Therefore the control goes to the else part and the output is BYE.

**4.What will be the output for the following program?**

|  |
| --- |
| public  class Test {  public      static void main(String[] args)      {          int a = 10, b = 20;     if (a < b) {              if (a > b) {                  System.out.println("HELLO GEEKS");              } else {                  System.out.println("WELCOME");              }          }      }  } |

Option:  
1.HELLO GEEKS  
2.WELCOME  
3.Compile time error  
4.HELLO GEEKS WELCOME

The answer is option (2)

**5. What will be the output for the following program?**

|  |
| --- |
| class Test {  public      static void main(String[] args)      {          for (int i = 0;; i++) {              System.out.println("HIII");          }          System.out.println("BYE");      }  } |

Options:  
1. HIII  
2. HIII(infinitely)  
3. BYE  
4. Compile time error

The answer is option (4)

**Explanation:** In the above for loop it will go for infinite loop and the above program does not give any chance to the next lines of the program. That’s why compiler will give compile time error saying error: unreachable statement.

**6.What is the output of the following question?**

|  |
| --- |
| class Test3 {  public      static void main(String[] args)      {          int x = 1;            if (x) {              System.out.print("GeeksForGeeks");          } else {              System.out.print("GFG");          }      }  } |

**Option**  
A) GeeksForGeeks  
B) GFG  
C) Error  
D) None

Output: C

**Explanation :**In Java, Compiler gives error – Incompatible types : int can not be converted to boolean type.  
**But in C or C++ its a valid statement.**

**7.what will be the output of the following program?**

|  |
| --- |
| public  class Test {  public      static void main(String[] args)      {          for (int i = 0; i < 10; i++)              int x = 10;      }  } |

Options:  
1. No Output  
2. 10  
3. Compile time error  
4. 10 (10 times)

The answer is option (3)

**Explanation:** Curly braces are optional and without curly braces we can take only one statement under for loop which should not be **declarative statement**. Here we are declaring a variable that’s why we will get compile time error saying error: variable declaration not allowed here.

**2. what will be the output of the following program?**

|  |
| --- |
| public  class Test {  public      static void main(String[] args)      {          for (int i = 0; String = "GFG"; i < 2; i++)              System.out.println("HELLO GEEKS"); //statement      }  } |

Options:  
1. HELLO GEEKS  
2. Compile time error  
3. HELLO GEEKS  
HELLO GEEKS  
HELLO GEEKS  
4. No Output

The answer is option (2)

**Explanation:** Initialization part of the for loop will be executed only once in the for loop life cycle. Here we can declare any number of variables but should be of same type. By mistake if we are trying to **declare different data types variables** then we will get compile time error saying error: incompatible types: String cannot be converted to int.

**3. what will be the output of the following program?**

|  |
| --- |
| public  class Test {  public      static void main(String[] args)      {          int i = 0;          for (System.out.println("HI"); i < 1; i++)              System.out.println("HELLO GEEKS");      }  } |

Output:  
1. HI  
HELLO GEEKS  
2. No Output  
3. Compile time error  
4. HELLO GEEKS

The answer is option (1)

**Explanation:**I n the initialization section we can take any valid java statement including System.out.println(). In the for loop **initialization section is executed only once** that’s why here it will print first HI and after that HELLO GEEKS

**4. what will be the output of the following program?**

|  |
| --- |
| public  class Test {  public      static void main(String[] args)      {          for (int i = 0;; i++)              System.out.println("HELLO GEEKS");      }  } |

Options:  
1. Compile time error  
2. HELLO GEEKS  
3. HELLO GEEKS (Infinitely)  
4. Run-time Exception

The answer is option (3)

**Explanation:**In the conditional check we can take any valid java statement but should be of type **Boolean**. If we did not give any statement then it always returns **true.**

**5. what will be the output of the following program?**

|  |
| --- |
| public  class Test {  public      static void main(String[] args)      {          for (int i = 0; i < 1; System.out.println("WELCOME"))              System.out.println("GEEKS");      }  } |

Options:  
1.GEEKS  
WELCOME  
GEEKS  
WELCOME  
2.No Output  
3.Compile time error  
4.GEEKS WELCOME(Infinitely)

Output:

The answer is option (4)

**Explanation:**In increment-decrement section we can take any valid java statement including System.out.println(). Here in the increment/decrement section, a statement is there, which result the program to go to infinite loop.

**While :**

**1**.What is the output of this program?

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {      unsigned int x = 3;      while (x-- >= 0) {          printf("%d  ", x);      }      return 0;  } |

**Option**  
a) 3 2 1 0  
b) 2 1 0 -1  
c) infinite loop  
d) -65535

Answer : C

**Explanation**: Here x is an unsigned integer and it can never become negative. So the expression x–>=0 will always be true, so its a infinite loop.

**Q.2** What is the output of this program?

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {      int x = 3, k;      while (x-- >= 0) {          printf("%d  ", x);      }      return 0;  } |

**option**:- a  
a) 3 2 1 0  
b) 2 1 0 -1  
c) infinite loop  
d) -65535

Answer: b

**Explanation:** Here x is an integer with value 3. Loop runs till x>=0 ; 2, 1, 0 will be printed and after x>=0, condition becomes true again and print -1 after false.

**Q.3** What is the output of this program?

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {      int x = 0, k;      while (+(+x--) != 0) {          x++;      }      printf("%d  ", x);      return 0;  } |

**option**  
a) 1  
b) 0  
c) -1  
d) infinite

Answer : C

**Explanation** Unary + is the only dummy operator in c++. So it has no effect on the expression.

**Q.4** What is the output of this program?

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {      int x = -10;      while (x++ != 0)          ;      printf("%d  ", x);      return 0;  } |

**option**  
a) 0  
b) 1  
c) -1  
d) infinite

Answer: b

**Explanation**: The semicolon is after the while loop. while the value of x become 0, it comes out of while loop. Due to post-increment on x the value of x while printing becomes 1.

**Q.5** What is the output of this program?

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {      while (1) {          if (printf("%d", printf("%d")))              break;          else              continue;      }      return 0;  } |

**option**  
a) Garbage value  
b) 1  
c) 0  
d) Error

Answer : a

**Explanation**: The inner printf executes and print some garbage value.

This article is contributed by **Gyayak Jain**. If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](http://www.contribute.geeksforgeeks.org/) or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

**Question 1**

|  |
| --- |
| #include"stdio.h"  int main()  {      int i;      if(i=0,2,3)          printf("Geeksforgeeks ");      else          printf("Hello ");      printf("%d\n",i);  } |

**OPTIONS:**  
a)Hello 3  
b)Hello 0  
c)Geeksforgeeks 0  
d)Geeksforgeeks 3

**OUTPUT:**  (c) Geeksforgeeks 0

**Explanation:**At first zero will assign in ‘i’ then comma operator returns the last value which is 3 and condition becomes true.

**Question 4**

|  |
| --- |
| #include"stdio.h"  int main()  {      int i;      if(i=(2,1,0))          printf("Geeksforgeeks ");      else          printf("Hello ");      printf("%d\n",i);  } |

**OPTIONS:**  
a)Hello 3  
b)Geeksforgeeks 0  
c)Hello 0  
d)Geeksforgeeks 3

**OUTPUT:**  (c) Hello 0

**Explanation:**Priority of parenthesis bracket is greater than equal to(=) operator , So atfirst comma operator return the last value which is zero(0) and then equal to(=) operator assign 0 to ‘i’ and condition becomes flase.

**Question 5**

|  |
| --- |
| #include"stdio.h"  int main()  {      float a=0.7d;      if(a<0.7)           printf("C");          else           printf("C++");      return 0;  } |

**OPTIONS:**  
a)Compilation error  
b)C++  
c)C

**OUTPUT:**  (C)C

**Explanation:**  
a = 0.7 is rounded to 0.699999988  
and the constant 0.7 is as 0.69999999999  
so a<0.7 is true so it print "c"  
but in case of 0.8  
a = 0.800000011 and  
constant 0.8 is 0.8000000000000000

**1.What will be the output of the following program?**

|  |
| --- |
| #include <stdio.h>  int main()  {      int x = 30, y = 25, z = 20, w = 10;      printf("%d ", x \* y / z - w);      printf("%d", x \* y / (z - w));      return (0);  } |

Options:  
1. 27 85  
2. 82 27  
3. 27 75  
4. No output

The output is option(3).

**Explanation :** In the above program, precedence takes the advantage. Therefore In the statement compiler first calculate x\*y then / and after that subtraction.

**3. Guess the output?**

|  |
| --- |
| #include <stdio.h>  int main()  {      int a = 15, b;      b = (a++) + (a++);      a = (b++) + (b++);      printf("a=%d b=%d", a, b);      return (0);  } |

Options:  
1. a=63 b=33  
2. a=33 b=63  
3. a=66 b=33  
4. a=33 b=33

The answer is option(1).

**Explanation:**Here,

a = 15 and b = (a++)+(a++) i.e. b = 15+16 = 31 and a =3 2.

Now a = (b++) + (b++) = 31 + 32 = 63 and b = 33.

Therefore the result is a = 63 and b = 33.

**4. What will be the output?**

|  |
| --- |
| #include <stdio.h>  int main()  {      main();      return (0);  } |

Options:  
1. compile time error  
2. abnormal termination  
3. run time error  
4. No output

The answer is option(1).

**Explanation:**Here stack overflow occurs that’s result in run-time error.

**1.What will be the output of following program?**

|  |
| --- |
| #include<stdio.h>  void main()  {      int i = 10;      static int x = i;      if (x == i)          printf("equal");      else if (x < i)))          printf("less than");      else          printf("greater than");  } |

**Options:**  
(a)equal  
(b)greater than  
(c)less than  
(d)compiler error  
(e)none of the above

**Answer: (d)Compiler error**

**Explanation:** Here ‘x’ is a [**static**](https://www.geeksforgeeks.org/static-variables-in-c/) variable and ‘i’ is an [**auto**](https://www.geeksforgeeks.org/type-inference-in-c-auto-and-decltype/) variable. Auto variables are run time entities, compared to static, which are load time entities. Run time variables cannot be initialized with load time variables.

**2. What will be the output of following program?**

|  |
| --- |
| #include<stdio.h>  void main()  {      printf("%s", "i"                   "am"                   "good");  } |

**Options:**  
(a)i am good  
(b)i  
(c)good  
(d)iamgood  
(e)Compiler error

**Answer: (d)iamgood**

**Explanation:** In C, string constant **“ab”** is same as **“a” “b”**.

**3. What will be the output of following program?**

|  |
| --- |
| #include<stdio.h>  #include <string.h>  void main()  {      printf("%d %d", sizeof("program"), strlen("program"));  } |

**Options:**

(a)7 7  
(b)8 8  
(c)8 7  
(d)7 8  
(e)None of the above

**Answer: (c)8 7**

**Explanation:** **[strlen](https://www.geeksforgeeks.org/difference-strlen-sizeof-string-c-reviewed/)** returns length of string without counting the null character, whereas **sizeof** also includes the null character when counting size of string.

**4. What will be the output of following program?**

|  |
| --- |
| #include<stdio.h>  void main()  {      int colour = 2;      switch (colour) {      case 0:          printf("Black");      case 1:          printf("Red");      case 2:          printf("Aqua");      case 3:          printf("Green");      default:          printf("Other");      }  } |

**Options:**  
(a)Aqua  
(b)AquaGreenOther  
(c)AquaGreen  
(d)Red  
(e)none of the above

.

**Answer: (b)AquaGreenOther**

**Explanation:** There are no **break** statements in [switch case](https://www.geeksforgeeks.org/switch-statement-cc/), so all statements after case 2, including the **default** statement, will be executed.

**5. What will be the output of following program?**

|  |
| --- |
| #include<stdio.h>  void main()  {      if (printf("cisgood"))          printf("i know c");      else          printf("i know c++");  } |

**Options:**  
(a)i know c  
(b)i know c++  
(c)cisgoodi know c  
(d)cisgoodi know c++  
(e)compiler error

**Answer: (c)cisgoodi know c**

**Explanation:** The return type of printf is **integer**, that is the number of characters including blank spaces. So, here in if condition, printf evaluates to 7, which is non-negative. Thus follows the true condition.