Predict the output of the following code snippets. Do not use any compiler. Do only Dry Run.

1. Predict the output of the following code snippets:

#include<iostream>

using namespace std;

class Base1 {

 public:

     Base1(){

cout << " Base1's constructor called" << endl;

}

};

class Base2 {

 public:

     Base2(){

cout << "Base2's constructor called" << endl;

}

};

class Derived: public Base1, public Base2 {

   public:

     Derived(){

  cout << "Derived's constructor called" << endl;

}

};

int main()

{

   Derived d;

   return 0;

}

Options:

1. Compiler Dependent
2. Base1’s constructor called

Base2’s constructor called

Derived’s constructor called

1. Base2’s constructor called

Base1’s constructor called

Derived’s constructor called

1. Compilation Error

**Solution: B**

**Explanation: Since the subclass (Derived class) inherited from two super classes base 1 and base 2(multiple inheritance).The execution order is Base 1 , Base 2 and Derived class are called**.

---------------------------------------------------------------------------------------------------------------------------

1. Predict the output of the following code snippets:

#include<iostream>

using namespace std;

class P {

public:

   void print()  { cout <<" Inside P"; }

};

class Q : public P {

public:

   void print() { cout <<" Inside Q"; }

};

class R: public Q { };

int main(void)

{

  R r;

  r.print();

  return 0;

}

Options:

1. Inside P
2. Inside Q
3. Compilation Error
4. Program will run without output

**Solution: B**

**Explanation: In the above code multilevel inheritance is seen i.e. class Q is the subclass of class P but it is the superclass to Class R (subclass of class Q). Since we create an object of class R it only inherits from class Q. So it prints only “Inside Q”.**

----------------------------------------------------------------------------------------------------------------------------

1. Predict the output of the following code snippets:
2. #include<iostream>
3. using namespace std;
4. class Base {};
5. class Derived: public Base {};
6. int main()
7. {
8. Base \*bp = new Derived;
9. Derived \*dp = new Base;
10. }

Options:

1. No Compilation Error
2. Runtime Error
3. Compilation Error in line 7
4. Compilation Error in line 8

**Solution: D**

**Explanation: A base class object cannot point to object of derived class**

----------------------------------------------------------------------------------------------------------------------------

1. Predict the output of the following code snippets:
2. #include<iostream>
3. using namespace std;
4. class Base
5. {
6. public:
7. void show()
8. {
9. cout<<" In Base ";
10. }
11. };
12. class Derived: public Base
13. {
14. public:
15. int x;
16. void show()
17. {
18. cout<<"In Derived ";
19. }
20. Derived()
21. {
22. x = 10;
23. }
24. };
25. int main(void)
26. {
27. Base \*bp, b;
28. Derived d;
29. bp = &d;
30. bp->show();
31. cout << bp->x;
32. return 0;
33. }

Options:

1. In Base 10
2. In Derived 10
3. Compilation Error in line 30
4. Compilation Error in line 31

**Solution: C**

**Explanation: Since we are creating an Object (bp) pointing to the base class it cannot be used to point the member variable(x) corresponding to the Derived class**

**But if we change line 26 and 27 (line 31 remains the same) such as**

Base b;

Derived \*bp ,d;

**We get the output as “In Derived 10”**

----------------------------------------------------------------------------------------------------------------------------

1. Predict the output of the following code snippets:

#include <iostream>

using namespace std;

class Animal

{

public:

int legs = 4;

};

class Dog : public Animal

{

public:

int tail = 1;

};

int main()

{

Dog d;

cout << d.legs;

cout << d.tail;

}

Options:

1. Error
2. 44
3. 40
4. 41

**Solution: D**

**Explanation:**

**Super class (Animal): legs=4**

**Subclass inherits from super class**

**Subclass (Dog): tail=1, legs=4**

**Subclass inherits from super class**

**Since we are creating object (d) of subclass (Dog) has both tail variable (1) and legs variable (4)(inherited from superclass animal)**

**Prints output as “41”**

----------------------------------------------------------------------------------------------------------------------------

1. Predict the output of the following code snippets:

#include <iostream>

using namespace std;

int main() {

int i = 0, x = 0;

do{

if(i % 5 == 0) {

cout<<x;

x++;

}

++i;

}while(i<10);

cout<<x;

return 0;

}

Options:

1. 01
2. 012
3. 0
4. 0123

**Solution: B**

**Explanation:**

|  |  |  |
| --- | --- | --- |
| **I value** | **X value** | **Output(at each iteration)** |
| **0** | **0** | **-** |
| **1** | **1** | **0** |
| **2** | **1** | **-** |
| **3** | **1** | **-** |
| **4** | **1** | **-** |
| **5** | **2** | **01** |
| **6** | **2** | **-** |
| **7** | **2** | **-** |
| **8** | **2** | **-** |
| **9** | **2** | **-** |

**Printing final x value (2) at the end gives the output as 012**

-------------------------------------------------------------------------------------------------------------------------

1. Predict the output of the following code snippets:

#include <iostream>

using namespace std;

int main() {

int i=0,x=0;

for(i=1;i<10;i\*=2){

x++;

cout<<x;

}

cout<<x;

return 0;

}

Options:

1. 1234567899
2. 12345678910
3. 123455
4. 12344

**Solution: D**

**Explanation (Dry Run):**

|  |  |  |
| --- | --- | --- |
| **I value** | **X value** | **Output at each stage:** |
| **0** | **0** | **-** |
| **1** | **1** | **1** |
| **2** | **2** | **12** |
| **4** | **3** | **123** |
| **8** | **4** | **1234** |

**Printing final x value (4) at the end gives the output as 12344**

-------------------------------------------------------------------------

1. How many times 'its a while loop' should be printed?

#include <iostream>

using namespace std;

int main(){

int i = 1 ;

i = i - 1 ;

while(i){

cout<<"its a while loop";

i++ ;

}

return 0;

}

Options:

1. 1
2. 2
3. 0
4. Infinite Times

**Solution: C**

**Explanation: Only if “I =1 or any non-zero value (true)”it will run infinite times else if I =0(false) it will not execute**

--------------------------------------------------------------------------------------------------------------------------------

1. What should be the output of below program?

#include <iostream>

using namespace std;

int main(){

int a = 1;

switch(a) {

case 1: cout<<"One";

case 2: cout<<"Two";

case 3: cout<<"Three";

default: cout<<"Default";

}

return 0;

}

Options:

1. One
2. Compilation Error
3. Default
4. OneTwoThree

**Solution: D**

**Explanation: Since the break statement is not present in the switch case code it switches to all the other cases in and print all the other statements atleast once also so the answer is D**

-------------------------------------------------------------------------------------------------------------------------------

1. What should be output of below program

if use enter a = 5?

#include <iostream>

using namespace std;

int main(){

int a;

cin>>a; // user can enter any value

if (++a\*5 <= 25) {

cout<<"Hello";

}

else {

cout<<"Bye";

}

}

Options:

1. Hello
2. Bye
3. Undefined
4. Compilation Error

**Solution: B**

|  |  |  |  |
| --- | --- | --- | --- |
| **a** | **++a\*5** | **++a\*5 <= 25** | **Output** |
| **5** | **30** | **False** | **Prints “Bye”** |

-----------------------------------------------------------------------------------------------------------------------------