OOP Test

What is the Output of the Program? Justify your answer.

1
```
#include <iostream>

using namespace std;
class base
{
   int a;
   base(){
      a=10;
   }
  void display_base()
   {
      cout<<a;
   }
};
class derived : private base
{
   int b;
   derived()
   {
      a=30;
      b=20;
   }
   void display_derived()
     {
      cout<<a;
      cout<<b;
  }
};

int main()
{
  base bob;
  derived dob;
  dob.display_derived();
  return 0;
}
```

2
```
#include <iostream>

using namespace std;
class base
{
   int a;
   base(){
      a=10;
   }
   void display_base()
   {
      cout<<a;
   }
 };
 class derived : public base
 {
    int b;
    derived()
    {
       a=30;
       b=20;
    }
    void display_derived()
       {
        cout<<a;
        cout<<b;
       }
  };

  int main()
  {
     base bob;
     derived dob;
     dob.display_derived();
     return 0;
  }
```

3

```
#include <iostream>

using namespace std;
class base
{
   int a;
   public:
   base(){
      a=10;
   }
  void display_base()
  {
     cout<<a;
  }
};
class derived : public base
{
   int b;
   public:
   derived()
   {
     a=30;
     b=20;
  }
  void display_derived()
     {
       cout<<a;
       cout<<b;
   }
};

int main()
{
   base bob;
   derived dob;
   dob.display_derived();
   return 0;
}
```

4

```
#include <iostream>

using namespace std;
class base
{
  int a;
  public:
  base(){
     a=10;
  }
  void display_base()
  {
     cout<<a;
  }
};
class derived : private base
{
   int b;
   public:
   derived()
   {
      a=30;
      b=20;
   }
   void display_derived()
     {
       cout<<a;
       cout<<b;
   }
};

int main()
{
   base bob;
   derived dob;
   dob.display_derived();
   return 0;
}
```

5

```
#include <iostream>

using namespace std;
class base
{
  protected:
  int a;
  public:
  base(){
    a=10;
  }
  void display_base()
  {
    cout<<a;
  }
};
class derived : private base
{
  int b;
  public:
  derived()
  {
    a=30;
    b=20;
  }
  void display_derived()
  {
    cout<<a;
    cout<<b;
  }
};

int main()
{
  base bob;
  derived dob;
  dob.display_derived();
  return 0;
}
```

6

```
#include <iostream>

using namespace std;
class base
{
  int a;
};
class derived : private base
{

};

int main()
{
  base bob;
  derived dob;
  cout<<sizeof(bob);
  cout<<sizeof(dob);
  return 0;
}
```

7

```
#include <iostream>

using namespace std;
class base
{
  int a;
};
class derived : public base
{
  int b;
};

int main()
{
  base bob;
  derived dob;
  cout<<sizeof(bob);
  cout<<sizeof(dob);
  return 0;
}
```

8

```
#include <iostream>

using namespace std;
class base
{
  protected:
  int a;
  public:
  base(){
    a=10;
  }
};
class derived : public base
{
  int b;
  public:
  derived(){
    b=20;
  }
  void display()
  {
    cout<<a<<b;
  }
};

int main()
{
   base bob;
   derived dob;
   dob.display();
   return 0;
}
```