PPL QUIZ 1 (21-02-2022)

u19cs012@coed.svnit.ac.in Switch account

Draft saved

Your email will be recorded when you submit this form

Quiz

Polymorphism in C++ requires:

○ Inheritance only

○ Virtual functions only

○ References only

● Inheritance, Virtual functions and references

Clear selection

Which of the following statements regarding inline functions and macros are incorrect?

○ Inline functions do argument type checking, whereas macros do not.

○ Macros are processed by the pre-processor, while inline functions are handled later in the compilation process.

○ Return statements are not allowed in macros, but they are allowed in inline functions.

● Inline functions are not as prone to flaws and errors as macros are.

○ None of these

Clear selection

By making both constructor and destructor private we can design a C++ class whose objects can only be created with the new operator. The application generates a compiler error if the user attempts to create an object directly.

● True

○ False

Clear selection

Which of the following statement concerning C++ references is FALSE?

● References cannot refer to constant value

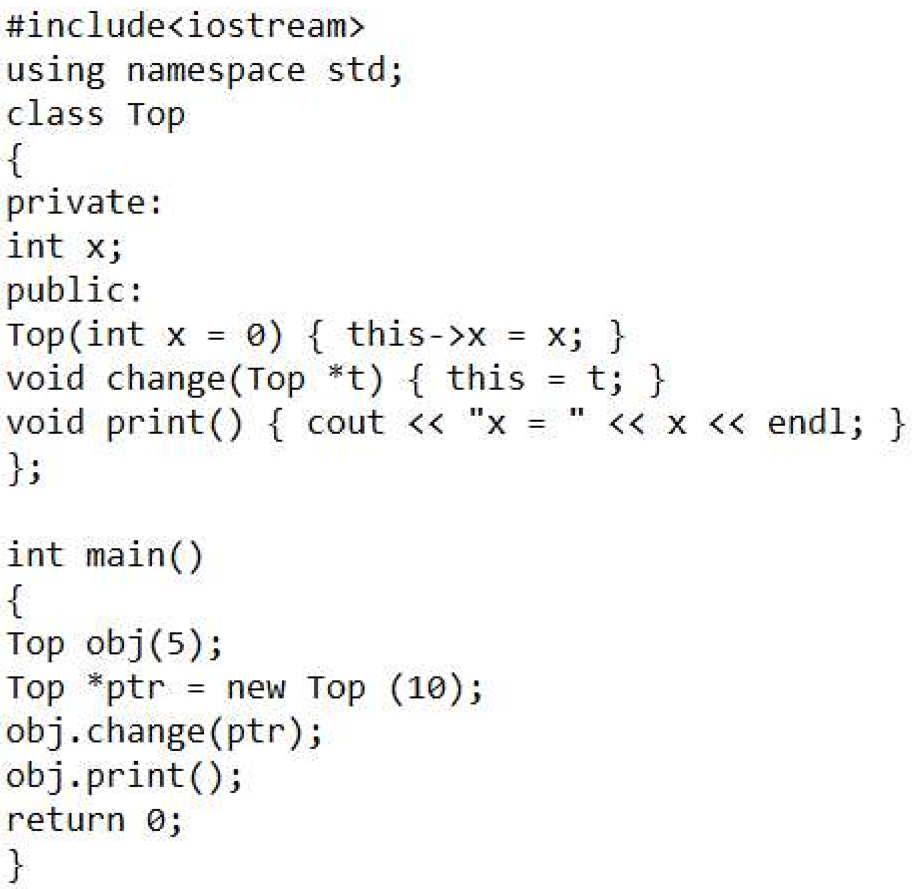
○ Once a reference is created, it cannot be later made to reference another object; it cannot be reset.

○ Reference must be initialized when declared

○ References cannot be NULL

Clear selection

Give the correct output of the following program.



```
#include<iostream>
using namespace std;
class Top
{
private:
int x;
public:
Top(int x = 0) { this->x = x; }
void change(Top *t) { this = t; }
void print() { cout << "x = " << x << endl; }
};

int main()
{
Top obj(5);
Top *ptr = new Top (10);
obj.change(ptr);
obj.print();
return 0;
}
```

● Compiler Error

○ x = 5

○ Runtime Error

○ x = 10

Clear selection

How to create a dynamic array of pointers (to integers) of size 10 using new in C++?

○ int *arr = new int *[10];

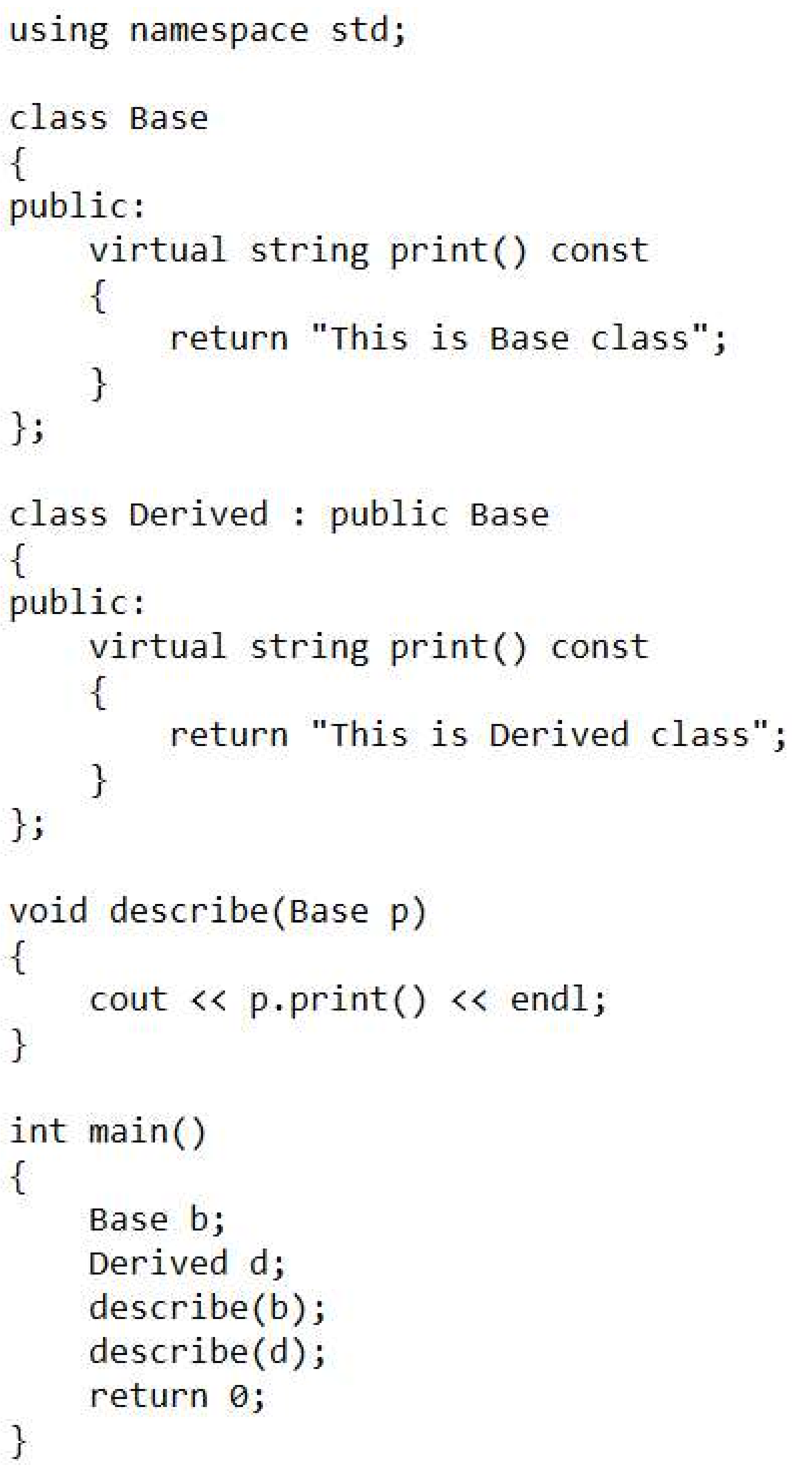
● int **arr = new int *[10];

○ int *arr = new int [10];

○ Not Possible

Clear selection

Output of following program?



```
using namespace std;

class Base
{
public:
    virtual string print() const
    {
        return "This is Base class";
    }
};

class Derived : public Base
{
public:
    virtual string print() const
    {
        return "This is Derived class";
    }
};

void describe(Base p)
{
    cout << p.print() << endl;
}

int main()
{
    Base b;
    Derived d;
    describe(b);
    describe(d);
    return 0;
}
```

○ This is Derived class This is Base class

○ This is Base class This is Derived class

○ Compilation Error

● This is Base class This is Base class

Clear selection

Which of the following is true about templates.

☑ Template is a feature of C++ that allows us to write one code for different data types.

☑ We can write one function that can be used for all data types including user defined types. Like sort(), max(), min(), ..etc.

☑ We can write one class or struct that can be used for all data types including user defined types. Like Linked List, Stack, Queue ..etc.

☑ Template is not an example of compile time polymorphism.

Which of the following is not correct (in C++) ?

□ Class templates and function templates are instantiated in the same way

☑ Class templates differ from function templates in the way they are initiated

☑ Class template is initiated by defining an object using the template argument

☑ Class templates are generally used for storage classes

Pure virtual functions -> which of the following statements is true?

□ None

☑ If a class has a pure virtual function, then the class becomes abstract class and an instance of this class cannot be created.

☑ Pure virtual function is implemented in derived classes

□ pure virtual return_type func();

□ Pure keyword should be used to declare a pure virtual function

Back Submit Clear form

Never submit passwords through Google Forms.