CCEE Mock - I | C++

Total points 26/40

The respondent's email (**amolgavit158121@gmail.com**) was recorded on submission of this form.

0 of 0 points

Name *

Amol Gavit

PRN (12 Digits) *

250240320013

Centre * Dropdown

Kharghar

Questions 26 of 40 points

✗ Identify the true statements about polymorphism, as implemented in C++. *0/1

i)Polymorphism allows objects of different classes that are related by inheritance to respond differently to the same member function call.
ii)Early binding is the mechanism for implementing polymorphism
iii)C++ supports polymorphism using virtual functions
iv)Polymorphism does not allow overloading of base-class member functions in derived classes.

- ○ i, iii and iv
- ○ ii, iii and iv
- ◉ Only iii ✗
- ○ i, ii and iii

Correct answer

- ◉ i, ii and iii

✓ Which of the following can be virtual? * 1/1

- ○ constructors
- ◉ destructors ✓
- ○ static functions
- ○ None of the above

What will be the output *

1/1

```
#include < iostream.h >
void main()
{
int arr[]={10,20,30,40,50};
int x,*ptr1=arr,*ptr2=&arr[3];
x=ptr2-ptr1;
cout << x;

}
```

- 6
- 3 ✓
- Compile Time Error
- Runtime Error

✓ Identify the true statements about memory allocation.? *1/1

i)A compiler allocates memory for a variable only if the new keyword is used?

ii)You must assign the memory address returned from a new statement to a pointer of the same type as the dynamically created variable.?

iii)The delete keyword can be used to free memory only if that memory was dynamically allocated using the new keyword.?

iv)When a statement that dynamically creates a variable is executed, the amount of memory allocated is determined automatically based on that variable`s type.

- ○ i, ii and iv
- ◉ iii and iv ✓
- ○ i, ii and iii
- ○ i and iii

✓ In protected inheritance: * 1/1

- ○ The public members of the base class become public.
- ◉ The public members of the base class become protected. ✓
- ○ The protected members of the base class become private.
- ○ The public members of the base class become inaccessible.

✓ Which of the following are not inherited * 1/1

- ○ Constructor functions
- ○ Overloaded constructors
- ○ Friend functions
- ◉ All of the given ✓

✓ A _____ is a "generic" function that can work with any data type * 1/1

- ○ function argument
- ○ function parameter
- ◉ function template ✓
- ○ None of these

✓ Which of the following keywords do you think can be used when declaring static members in a class? *1/1
(i) Public
(ii) Private
(iii) Protected

- ◉ i, ii and iii. ✓
- ○ i and ii.
- ○ Only i.
- ○ i and iii.

✓ A null pointer is a pointer that contains * 1/1

- (●) the address 0 ✓
- ( ) the address that points to 0
- ( ) the address that points to '\0'
- ( ) the address that points to -1

✓ Which choice does not produce the same output as this code snippet? Assume the variable i will not be used anywhere else in the code. *1/1

```
for (i=1;i<10;i++){
   cout<<i<<endl;
}
```

A.
```
i=1;
while(i<10){
   cout<<++i<<endl;
}
```

B.
```
for (int i:{1,2,3,4,5,6,7,8,9}) {
   cout<<i<<endl;
}
```

C.
```
i = 1;
do {
   cout<<i++<<endl;
} while(i<10);
```

D.
```
i = 1;
loop:
   cout<<i++<<endl;
   if(i<10) goto loop;
```

- (●) A ✓
- ( ) B
- ( ) C
- ( ) D

✓ What is the output of this code? * 1/1

```
#include <iostream>

int main(){
   int x=10, y=20;
   std::cout << "x = " << x++ << " and y = " << --y << std::endl;
   std::cout << "x = " << x-- << " and y = " << ++y << std::endl;
   return(0);
}
```

- ○ x = 10 and y = 20 x = 11 and y = 19
- ○ x = 11 and y = 19 x = 10 and y = 20
- ◉ x = 10 and y = 19 x = 11 and y = 20 ✓
- ○ x = 11 and y = 20 x = 10 and y = 19

✓ Which of the following is not a difference between a class and a struct? * 1/1

- ○ Because structs are part of the C programming language, there are some complexity between C and C++ structs. This is not the case with classes.
- ◉ Classes may have member functions; structs are private. ✓
- ○ The default access specifier for members of struct is public, whereas for member of class, it is private.
- ○ Template type parameters can be declared with classes, but not with the struct keyword.

✓ It is possible to take address of member of class and assign to a pointer *1/1

- (●) True ✓
- ( ) False
- ( ) Unka address miljaye bas
- ( ) Life set hojaegi

✗ what will be the C++ output of the following code * 0/1

```
#include < iostream.h >
void fooling(int i,int &j,int k)
{
return(i=j=k);
}
void fooling_result(int i)
{
cout << i;
}
main()
{
int j=3;
int x=fooling(1,j,4)
fooling_result(x);
result(0);
}
```

- ○ 6
- ◉ 10 ✗
- ○ error
- ○ 5

Correct answer

- ◉ error

✗ Identify the true statements about function templates. *0/1

i)All template definitions must be preceded by the keyword `class`
ii)The code in a template changes every time the function template is instantiated
iii)Every formal parameter in the template definition must appear at least once in the function`s parameter list.
iv)Formal parameter names must be unique in the parameter list of a template function

- ○ iii and iv
- ○ ii, iii and iv
- ◉ i, ii, iii and iv ✗
- ○ i and ii

Correct answer

- ◉ iii and iv

✓ Data members of a class can be qualified as static * 1/1

- ◉ True ✓
- ○ False
- ○ Only static friends
- ○ Only void static

✓ What is a destructor? * 1/1

- ○ A function called when an instance of a class is initialized.
- ◉ A function that is called when an instance of a class is deleted. ✓
- ○ A special function to change the value of dynamically allocated memory.
- ○ A function that is called in order to change the value of a variable

✗ what will be c++ output following code * 0/1

```
main()
{
BASE B_object;
BASE *B_ptr;
DERIVED1 D1_object;
DERIVED1 *D1_ptr;
B_object.BB=10;
D1_object.BB=30;
B_ptr=&D1_object;
B_ptr->BB=20;
B_ptr = &B_object;
B_ptr->disp();
}
```

- ◉ 20 ✗
- ○ 10
- ○ 30
- ○ compilation error

Correct answer

- ◉ 10

✓ To delete a dynamically allocated array named ‘a’ , the correct statement is *1/1

- ○ delete a;
- ○ delete a[0];
- ◉ delete []a; ✓
- ○ delete [0]a;

✓ Which is False ? * 1/1

- ○ Multilevel inheritance happens when a derived class becomes a base class
- ○ Hierarchical inheritance leads to multiple derived classes from a base class
- ◉ A derived class can not be used as a base class ✓
- ○ Hybrid inheritance is a special case of multiple inheritance

✓ What is the ternary operator equivalent to this code snippet? * 1/1

```
if(x)
   y=a;
else
   y=b;
```

- ○ y=a?b:x;
- ○ y=if(x?a:b);
- ○ y=(x&a)?a:(x&b)?b:0;
- ◉ y=x?a:b; ✓

✗ Which of the following do you think happens if the throw() function has an empty exception specification, and you place it after a function`s parameter list? *0/1

- ○ No exception is thrown.
- ○ A default exception is thrown.
- ○ The first exception encountered is thrown.
- ◉ Program will have an unexpected behavior. ✗

Correct answer

- ◉ No exception is thrown.

✗ In case of a copy constructor, which of the following is true? * 0/1

(a) Used to instantiate an object from another existing object
(b) To copy one object to another existing object.
(c) Can be a substitute for a '=' operator.

- (●) Both A and B ✗
- ( ) All of the mentioned
- ( ) Only A
- ( ) Only C

Correct answer

- (●) All of the mentioned

✓ Virtual functions * 1/1

- ( ) Must be static member of base class.
- ( ) Must be a static member of base class which must be defined.
- (●) Must be a non static member ✓
- ( ) Must be static member of base class which need not be defined.

✗ In C++, only one catch block can handle all the exceptions. * 0/1

- ○ True
- ◉ False ✗

Correct answer

- ◉ True

✓ When the compiler binds a member function call with the version of the function that resides in same class as the itself, this is considered_____binding. *1/1

- ○ local
- ○ safe
- ◉ static ✓
- ○ dynamic

✓ During inheritance which of the following are not inherited. * 1/1

- ○ friends
- ○ constructors
- ○ assignment operator
- ◉ All the mentioned ✓

✓ The base class`s _____affects the way its members are inherited by the derived class. *1/1

- ○ name
- ○ return data type
- ◉ access specification ✓
- ○ a and b

✗ An array with no elements is * 0/1

- ◉ legal in c++ ✗
- ○ illegal in C++
- ○ automatically furnished one element,with a value of zero
- ○ automatically furnished one value -- the null terminator

Correct answer

- ◉ illegal in C++

✗ Identify the true statements about abstract classes. *0/1

i)You cannot create a pointer to an abstract base class.
ii)Abstract base classes define real objects.
iii)You can have abstract classes at more than one layer of a hierarchy.
iv)If a derived class fails to redefine a pure virtual member function of the base class, then the derived class becomes an abstract class.

- ○ All of the above.
- ○ i, iii and iv
- ◉ ii, iii and iv ✗
- ○ iii and iv

Correct answer

- ◉ iii and iv

✓ Which of the following keywords do you think can be used when declaring static members in a class? *1/1

i)Public
ii)Private
iii)Protected

- ○ Only i
- ○ i and ii
- ◉ i, ii and iii ✓
- ○ i and iii

✗ Which of the following is a true statement about the difference between pointers and iterators? *0/1

- ○ While pointers are variable that hold memory address, iterators are generic functions used to traverse containers. These function allows the programmer to implement read and write code as the container is traversed.
- ○ Incrementing an iterator always means access the next element in the container(if any), no matter the container. Incrementing the pointer means pointing to the next element in memory, not always the next element.
- ◉ Pointers are variables that hold memory address where as iterator are unsigned integers that refers to offsets in arrays. ✗
- ○ All iterator are implemented with pointers so all iterators are pointers but not all pointers are iterators.

Correct answer

- ◉ Incrementing an iterator always means access the next element in the container(if any), no matter the container. Incrementing the pointer means pointing to the next element in memory, not always the next element.

✓ What would be the output of this code? * 1/1

```
int i0=4, i1=6, i2=8;
int& nums[3]={i2,i0,i1};
std::cout<<nums[0]<<nums[1]<<nums[2];
```

- ◉ compiler error ✓
- ○ 846
- ○ The output is the addresses of i2, i0, and i1, in that order, with no spaces.
- ○ 468

✗ The dynamic_cast operator * 0/1

- ○ allows only upcasting in the class hierarchy
- ◉ allows only downcasting in the class hierarchy ✗
- ○ allows both upcasting and downcasting
- ○ None

Correct answer

- ◉ allows only upcasting in the class hierarchy

✓ Which of the following statement is FALSE about pointers? * 1/1

- ○ The ++ and -- operators may be used with pointer variables
- ○ An integer may be added and subtracted from a pointer variable
- ◉ A pointer may be added to another pointer. ✓
- ○ A pointer may be subtracted from another pointer.

✓ To set object values which is more efficient * 1/1

- ○ assignment
- ○ use of = operator
- ◉ initialization ✓
- ○ any of the mentioned

✓ The only difference between the get function and the >> operator is that get reads the first character typed , even if it is a space , tab, or the [Enter] key . *1/1

- (●) True ✓
- ( ) False

✗ What is wrong with the following program? * 0/1

```
#include<iostream.h>
void main()
{
do
{
int b=0;
cout<<b;
b++;
}while(b!=10);
}
```

- ( ) There is nothing wrong in the program.
- (●) Variable ‘b’ must not be initialized in the loop ✗
- ( ) Variable ‘b’ must not be declared in the loop
- ( ) The condition for while loop is not valid

Correct answer

- (●) Variable ‘b’ must not be declared in the loop

✗ Friend Function * 0/1

- ○ is declared as friend and defined at the same time
- ◉ is always an inline function ✗
- ○ creates objects with initialization
- ○ None

Correct answer

- ◉ None

✗ what will be the output of the following code * 0/1

```
#include < iostream.h >
main()
{
int i1;
cout << endl << "Enter a 4 digit Integer:";
cin >> i1;
try
{
if(i1 < 1000)
throw(1);
else
{
if(i1%2)
throw(2);
else
throw(3);
}
}
catch(int i1)
{
if(i1==1)
cout << "Number less than 1000";
else
if(i1==2)
cout << "even number";
else
cout << "odd number";
}
return(0);
}
```

- (●) Number less than 1000 ✗
- ( ) compliation error
- ( ) Even number
- ( ) Oddnumber

Correct answer

- (●) Oddnumber

Feedback 0 of 0 points

From now onwards, I will give my best in everything in my life without any excuses. Because I know, problem is the part and parcel of life. We should always look for solutions. *

- (●) I PROMISE

How was your experience ? Anything you want to share?

Difficulty level of mock * Dropdown

Tough