“Abstract Class”.

1. Which among the following best describes abstract classes?
a) If a class has more than one virtual function, it’s abstract class
b) If a class have only one pure virtual function, it’s abstract class
c) If a class has at least one pure virtual function, it’s abstract class
d) If a class has all the pure virtual functions only, then it’s abstract class
View Answer
Answer: c
Explanation: The condition for a class to be called abstract class is that it must have at least one pure virtual function. The keyword abstract must be used while defining abstract class in java.

2. Can abstract class have main() function defined inside it?
a) Yes, depending on return type of main()
b) Yes, always
c) No, main must not be defined inside abstract class
d) No, because main() is not abstract function
View Answer
Answer: b
Explanation: This is a property of abstract class. It can define main() function inside it. There is no restriction on its definition and implementation.

3. If there is an abstract method in a class then, ________________
a) Class must be abstract class
b) Class may or may not be abstract class
c) Class is generic
d) Class must be public
View Answer
Answer: a
Explanation: It is a rule that if a class have even one abstract method, it must be an abstract class. If this rule was not made, the abstract methods would have got skipped to get defined in some places which are undesirable with the idea of abstract class.

4. If a class is extending/inheriting another abstract class having abstract method, then

____

a) Either implementation of method or making class abstract is mandatory
b) Implementation of the method in derived class is mandatory

c) Making the derived class also abstract is mandatory
d) It’s not mandatory to implement the abstract method of parent class
View Answer
Answer: a
Explanation: Either of the two things must be done, either implementation or declaration of class as abstract. This is done to ensure that the method intended to be defined by other classes gets defined at every possible class.

5. Abstract class A has 4 virtual functions. Abstract class B defines only 2 of those member functions as it extends class A. Class C extends class B and implements the other two member functions of class A. Choose the correct option below.
a) Program won’t run as all the methods are not defined by B
b) Program won’t run as C is not inheriting A directly
c) Program won’t run as multiple inheritance is used
d) Program runs correctly
View Answer
Answer: d
Explanation: The program runs correctly. This is because even class B is abstract so it’s not mandatory to define all the virtual functions. Class C is not abstract but all the virtual functions have been implemented will that class.

Check this: Computer Science Books | MCA MCQs

6. Abstract classes can ____________________ instances.
a) Never have
b) Always have
c) Have array of
d) Have pointer of
View Answer
Answer: a
Explanation: When an abstract class is defined, it won’t be having the implementation of at least one function. This will restrict the class to have any constructor. When the class doesn’t have constructor, there won’t be any instance of that class.

7. We ___________________ to an abstract class.
a) Can create pointers
b) Can create references
c) Can create pointers or references
d) Can’t create any reference, pointer or instance
View Answer
Answer: c
Explanation: Even though there can’t be any instance of abstract class. We can always

create pointer or reference to abstract class. The member functions which have some implementation inside abstract itself can be used with these references.

8. Which among the following is an important use of abstract classes?
a) Header files
b) Class Libraries
c) Class definitions
d) Class inheritance
View Answer
Answer: b
Explanation: The abstract classes can be used to create a generic, extensible class library that can be used by other programmers. This helps us to get some already implemented codes and functions that might have not been provided by the programming language itself.

9. Use of pointers or reference to an abstract class gives rise to which among the following feature?
a) Static Polymorphism
b) Runtime polymorphism
c) Compile time Polymorphism
d) Polymorphism within methods
View Answer
Answer: b
Explanation: The runtime polymorphism is supported by reference and pointer to an abstract class. This relies upon base class pointer and reference to select the proper virtual function.

10. The abstract classes in java can _________________
a) Implement constructors
b) Can't implement constructor
c) Can implement only unimplemented methods
d) Can't implement any type of constructor
View Answer
Answer: a
Explanation: The abstract classes in java can define a constructor. Even though instance can't be created. But in this way, only during constructor chaining, constructor can be called. When instance of concrete implementation class is created, it's known as constructor chaining.

11. Abstract class can't be final in java.
a) True

b) False
View Answer
Answer: a
Explanation: If an abstract class is made final in java, it will stop the abstract class from being extended. And if the class is not getting extended, there won’t be another class to implement the virtual functions. Due to this contradicting fact, it can’t be final in java.

12. Can abstract classes have static methods (Java)?
a) Yes, always
b) Yes, but depends on code
c) No, never
d) No, static members can’t have different values
View Answer
Answer: a
Explanation: There is no restriction on declaring static methods. The only condition is that the virtual functions must have some definition in the program.

13. It is ____________________ to have an abstract method.
a) Not mandatory for an static class
b) Not mandatory for a derived class
c) Not mandatory for an abstract class
d) Not mandatory for parent class
View Answer
Answer: c
Explanation: Derived, parent and static classes can’t have abstract method (We can’t say what type of these classes is). And for abstract class it’s not mandatory to have abstract method. But if any abstract method is there inside a class, then class must be abstract type.

14. How many abstract classes can a single program contain?
a) At most 1
b) At least 1
c) At most 127
d) As many as required
View Answer
Answer: d
Explanation: There is no restriction on the number of abstract classes that can be defined inside a single program. The programs can use as many abstract classes as required. But the functions with no body must be implemented.

15. Is it necessary that all the abstract methods must be defined from an abstract class?

a) Yes, depending on code
b) Yes, always
c) No, never
d) No, if function is not used, no definition is required
View Answer
Answer: b
Explanation: That is the rule of programming language that each function declared, must have some definition. There can't be some abstract method that remains undefined. Even if it's there, it would result in compile time error.

"Template Class".

1. A template class can have ______________
a) More than one generic data type
b) Only one generic data type
c) At most two data types
d) Only generic type of integers and not characters
View Answer
Answer: a
Explanation: The template class can support more than one data type. The only thing is to add all the data types required in a list separated by comma within template specification.

2. Which among the following is the proper syntax for the template class?
a) template <typename T1, typename T2>;
b) Template <typename T1, typename T2>;
c) template <typename T> T named(T x, T y){ }
d) Template <typename T1, typename T2> T1 named(T1 x, T2 y){ }
View Answer
Answer: c
Explanation: The syntax must start with keyword template, case sensitive. Then it should include the typename and a variable to denote it. Then whenever that variable is used, it replaces it with the data type needed.

3. Can default arguments be used with the template class?
a) Yes, in some special cases
b) Yes, always
c) No, it must satisfy some specific conditions first

d) No, it can't be done
View Answer
Answer: b
Explanation: The template class can use default arguments. This is used to specify the data type to be considered if it is not specified while passing to the generic class. The default type will be used.

4. What is the syntax to use explicit class specialization?
a) template <int> class myClass<>{ }
b) template <int> class myClass<int>{ }
c) template <> class myClass<>{ }
d) template <> class myClass<int>{ }
View Answer
Answer: d
Explanation: The class specialization is creation of explicit specialization of a generic class. We have to use template<> constructor for this to work. It works in the same way as with explicit function specialization.

5. Which is the most significant feature that arises by using template classes?
a) Code readability
b) Ease in coding
c) Code reusability
d) Modularity in code
View Answer
Answer: c
Explanation: The code reusability is the feature that becomes more powerful with the use of template classes. You can generate a single code that can be used in variety of programming situations.

6. A template class defines the form of a class ____________________ it will operate.
a) With full specification of the data on which
b) With full specification of the functions on which
c) Without full specification of the data on which
d) Without full specification of the functions on which
View Answer
Answer: c
Explanation: The template classes can accept all types of data types. There is no need

to specify the data on which the class has to operate. Hence it gives us flexibility to code without worrying about the type of data that might be used in the code.

7. What are the two specializations of I/O template classes in C++?
a) 16-bit character and wide characters
b) 8-bit character and wide characters
c) 32-bit character and locale characters
d) 64-bit characters and locale characters
View Answer
Answer: b
Explanation: The I/O specialization is made with wide character and 8-bit characters. Wide characters are used to store the characters that might take more than 1 byte of space in memory or any size that is different from the one that the machine is using.

8. Can typeid() function be used with the object of generic classes?
a) Yes, only if default type is given
b) Yes, always
c) No, generic data can't be determined
d) No, never possible
View Answer
Answer: b
Explanation: The typeid() function can be used with the objects of generic classes. An instance of a template class will take the type of data that is being used with it. Hence when typeid() function is used, the data type would have already been defined and hence we can get desired result from typeid() function.

9. The _____________ class is a specialization of a more general template class.
a) String
b) Integer
c) Digit
d) Math
View Answer
Answer: a
Explanation: The string class is more specialized. Since the string must be able to store any kind of data that is given to the string. Hence it needs maximum specialization.

10. How is function overloading different from template class?
a) Overloading is multiple function doing same operation, Template is multiple function doing different operations
b) Overloading is single function doing different operations, Template is multiple function doing different operations
c) Overloading is multiple function doing similar operation, Template is multiple function doing identical operations

d) Overloading is multiple function doing same operation, Template is same function doing different operations

View Answer

Answer: c

Explanation: The function overloading is multiple functions with similar or different functionality but generic class functions perform the same task on given different types of data.

11. What if static members are declared inside template classes?

a) All instances will share the static variable

b) All instances will have their own static variable

c) All the instances will ignore the static variable

d) Program gives compile time error

View Answer

Answer: b

Explanation: The generic class have a special case with static members. Each instance will have its own static member. The static members are not shared usually.

12. What is the output of following program?

```
template <typename T>
void test(const T&x)
{
    static int count = 0;
    cout &lt;&lt; "x = " &lt;&lt; x &lt;&lt; " count = " &lt;&lt; count &lt;&lt; endl;
    ++count;
    return;
}

void main()
{
    test<int> (2);
    test<int>(2);
    test<double>(2.2);
}
```

a)

```
x = 2 count = 0

x = 2.2 count = 0

x = 2.2 count = 0
```

b)

```
x = 2 count = 0

x = 2 count = 0

x = 2.2 count = 0
```

c)

```
x = 2 count = 0

x = 2 count = 1

x = 2.2 count = 0
```

d)

```
x = 2 count = 0

x = 2 count = 1

x = 2.2 count = 2
```

View Answer

Answer: c

Explanation: For each new type, the class will have separate instance. Here two instances will be created and hence counter for integer goes to 1. And for float value, the count remains 0 for the output.

13. If template class is defined, is it necessary to use different types of data for each call?

a) No, not necessary
b) No, but at least two types must be there
c) Yes, to make proper use of template
d) Yes, for code efficiency

View Answer

Answer: a

Explanation: It is not necessary to use different type with each call to the generic function. Data may be of same type with each call but still the function works. We don’t consider other properties like efficiency with this concept because it is made generic to all data type, hence always works.

14. How many generic types can be given inside a single template class?
a) Only 1
b) Only 3
c) Only 7
d) As many as required
View Answer
Answer: d
Explanation: There is no restriction on the number of types to be used for making the class generic. There can be any number of generic types with a single class. Hence giving flexibility to code with all the data types.

15. Template classes must have at least one static member.
a) True
b) False
View Answer
Answer: b
Explanation: There is no mandatory condition to have static members inside template class. Not only template, it is not mandatory to have static members anywhere. We can use them as required in the code.