Question 1

Which statement is true about classes and objects?

1.  Objects state the objectives of a class.
2.  Classes classify objects in separate groups.
3.  Classes describe the structure and behaviour of similar objects. An object is an instance of a class.
4.  Objects describe the structure and behaviour of similar classes. A class is an instance of an object.
5. You answered this question correctly.

Question 2

What statement is true about the following code?

Code (C++):

1. #include <iostream>
3. void Swap(int& a, int& b)
4. {
5. int tmp=a;
6. a=b;
7. b=tmp;
8. }
10. int main()
11. {
12. int i1=10;
13. int i2=20;
15. Swap(i1, i2);
17. std::cout<<"i1="<<i1<<", i2="<<i2<<std::endl;
18. }

1.  The parameters to the *Swap* function are passed by reference.
2.  The program does not compile.
3.  The parameters to the *Swap()* are passed by value.
4.  The parameters to the *Swap* function are passed as pointer.

Question 3

Which statement is false about classes and objects?

1.  Objects have state and behaviour
2.  Objects have state, behaviour and identity
3.  Classes have state, behaviour and identity
4.  Classes have state and behaviour

Question 4

What statement is false about constructors and destructors?

1.  A constructor initialises an object.
2.  A destructor is used to clean up resources.
3.  A destructor is automatically called when an object gets out of scope.
4.  You can have multiple constructors and multiple destructors.
5. You answered this question correctly.

Question 5

What statement is false about function name overloading?

1.  Two functions can have the same name as long as the number of input arguments are different.
2.  Two functions can have the same name as long as the input arguments have different types.
3.  Two functions can have the same name as long as the output arguments have different types.
4.  Overloading works with both member functions and global functions.
5. You answered this question correctly.

Question 6

What statement is true about the use of #ifndef/#define/#endif statements in a class header file?

Code (C++):

2. #ifndef MyClass\_hpp
3. #define MyClass\_hpp
5. class MyClass
6. {
7. };
9. #endif

1.  The name of the #define must be the same as the header file.
2.  The #indef/#define/#endif statements are needed to ensure the class declaration can only be included once in each compilation unit.
3.  The name of the #define must be the same as the class name.
4.  The #indef/#define/#endif statements are needed to make the class known to the compiler. Else other files can't find the class that is declared.
5. You answered this question correctly.

Question 7

What statement is true about the following code?

Code (C++):

1. class A;

1.  'A' is an empty class.
2.  'A' is a forward declaration. The body is implemented elsewhere.
3.  This code does not compile.
4.  'A' is a local variable of type *class*.

Question 8

We want to swap two doubles. Which of the following functions is the best and most user-friendly?

1.  void Swap(double& d1, double& d2);
2.  void Swap(double\* d1, double\* d2);
3.  void Swap(double d1, double d2);
4.  void Swap(const double& d1, const double& d2);
5. You answered this question correctly.

Question 9

Which statement is false about constructors?

1.  If we don't make a default constructor, then the system only creates one with a standard implementation when we didn't create any other constructors.
2.  If we don't make a default constructor, then the system always creates one with a standard implementation.
3.  If we don't make a copy constructor, then the system always creates one with a standard implementation.
4.  The copy constructor copies the state of an object and must accept a reference to the source object.

Question 10

Which statement is true about inheritance and aggregation?

1.  Aggregation specialises a general class.
2.  Aggregation is used when two classes have "is a kind of" (ISA/AKO) relation.
3.  Inheritance is used when two class have a "has a" relationship.
4.  Inheritance is used when two classes have "is a kind of" (ISA/AKO) relation.

Question 1

Which statement is false about classes and objects?

1.  Objects have state, behaviour and identity
2.  Objects have state and behaviour
3.  Classes have state and behaviour
4.  Classes have state, behaviour and identity
5. You answered this question correctly.

Question 2

What statement is false about function name overloading?

1.  Two functions can have the same name as long as the number of input arguments are different.
2.  Two functions can have the same name as long as the input arguments have different types.
3.  Overloading works with both member functions and global functions.
4.  Two functions can have the same name as long as the output arguments have different types.
5. You answered this question correctly.

Question 3

Which statement is true about data hiding?

1.  Data hiding saves memory space.
2.  Data hiding ensures the data cannot be changed.
3.  Data hiding is mandatory in C++.
4.  Data hiding hides the internal data of a class from users of the class so the internal structure can be changed without affecting the users of a class.
5. You answered this question correctly.

Question 4

What statement is false about "pass by value" vs. "pass by reference"?

1.  To pass an argument by reference you need to declare the input parameter with a '&'
2.  Pass by value is less efficient than pass by reference for objects.
3.  Pass by value makes a copy of the argument.
4.  To pass an argument by reference you need to declare the input parameter with a '\*'.
5. You answered this question correctly.

Question 5

What statement is true about the following class?

Code (C++):

2. // In the header file: "MyClass.hpp"
3. class MyClass
4. {
5. int m\_data1;
7. MyClass();
9. private:
10. double m\_data2;
11. }
13. // In the source file: "MyClass.cpp"
14. #include "MyClass.hpp"
15. MyClass::MyClass()
16. {
17. }

1.  The initial value of *m\_data2* is 0.0.
2.  This class declaration does not compile.
3.  *m\_data1* is a public data member.
4.  Client code can instantiate this class because it defines a default constructor.

Question 6

Which statement is true about inheritance and aggregation?

1.  Inheritance is used when two class have a "has a" relationship.
2.  Aggregation specialises a general class.
3.  Inheritance is used when two classes have "is a kind of" (ISA/AKO) relation.
4.  Aggregation is used when two classes have "is a kind of" (ISA/AKO) relation.
5. You answered this question correctly.

Question 7

What statement is true about the following code?

Code (C++):

1. // In the header file: "MyClass.hpp"
2. class MyClass
3. {
4. private:
5. double m\_data;
7. public:
8. MyClass(double data);
9. };
11. // In the source file: "MyClass.cpp"
12. #include "MyClass.hpp"
13. MyClass(double d)
14. {
15. m\_data=data;
16. }

1.  The class does not compile because the constructor with *double* argument is not implemented.
2.  The class does not compile since the constructor cannot initialise *m\_data* because it is private.
3.  The class does not compile because the constructor with double argument has no return argument.
4.  The class does not compile because there is no default constructor.

Question 8

We want to swap two doubles. Which of the following functions is the best and most user-friendly?

1.  void Swap(double d1, double d2);
2.  void Swap(const double& d1, const double& d2);
3.  void Swap(double\* d1, double\* d2);
4.  void Swap(double& d1, double& d2);
5. You answered this question correctly.

Question 9

What statement is true about the following code?

Code (C++):

1. class A;

1.  This code does not compile.
2.  'A' is an empty class.
3.  'A' is a forward declaration. The body is implemented elsewhere.
4.  'A' is a local variable of type *class*.
5. You answered this question correctly.

Question 10

Which of the following statements states three key features of object oriented programming?

1.  Functions, variables and classes.
2.  Classes, objects and variables.
3.  Data hiding, classes and encapsulation.
4.  Encapsulation, data hiding and inheritance.

Question 6

What is encapsulation in the context of object-oriented programming?

1.  Bundling data with functionality that operates on that data.
2.  Hiding data from users.
3.  The process of writing a class.
4.  Compiling multiple classes in to one executable file.

Question 1

Which statement is true about *const*?

1.  Const member functions make the current object state const during that function.
2.  Const member functions can't change any data.
3.  You cannot have a const and a non-const member function with the same name and input- and output-arguments.
4.  Only const variables can be passed to functions with const parameters.