Quiz 23 Section B

OOP Section-B Quiz
Note: There might be MCQs with more than 1 option correct you need to mark the option which you think is closest one.

The respondent's email (**bsse23021@itu.edu.pk**) was recorded on submission of this form.

Which of these features of OOP would indicate code reusability? * 1 point

- ○ Polymorphism
- ○ Abstraction
- ◉ Inheritance
- ○ Encapsulation

What would be output of following code snippet? * 1 point

```
#include <iostream>
using namespace std;

class Base {
public:
    virtual void show() { cout << "Base class\n"; }
};

class Derived: public Base {
public:
    void show() { cout << "Derived class\n"; }
};

int main() {
    Base* b;
    Derived d;
    b = d;
    b->show();
    return 0;
}
```

- (●) Error
- ( ) Derived Class
- ( ) Base Class
- ( ) No Output

Which of these specifiers would be applied to the constructors only? * 1 point

- ○ Implicit
- ○ Protected
- ○ Public
- ◉ Non of these

The access specifier that is/are the most secure during inheritance is/are * 1 point

- ○ Protected
- ◉ Private
- ○ Default
- ○ Default & Private

If in case, in multiple inheritances, a class R would inherit the Class Q, while Class Q would inherit the class P, then in which sequence would their destructors be called in case we declare an object of Class R? * 1 point

- ◉ ~R() then ~P() then ~Q()
- ○ ~P() then ~Q() then ~R()
- ○ ~Q() then ~R() then ~P()
- ○ ~R() then ~Q() then ~P()

What is the name of the feature in which we enforce the definitions of the abstract function at the compile time? * 1 point

- [ ] All of these
- [ ] Dynamic or Static Polymorphism according to need
- [x] Static Polymorphism
- [ ] Dynamic Polymorphism

* 1 point

Which of the following is not a type of inheritance?

```
class Car
{ public: virtual void start() = 0;  };
```

- [ ] Non of these
- [ ] Association
- [x] Encapsulation
- [ ] Abstraction

Copy constructor must receive its arguments by? * 1 point

- [ ] pass by value
- [ ] pass by address
- [x] pass by reference
- [ ] pass by reference or pass by address

What is the output of the following C++ code if the input is "9 14"? * 1 point

```
#include <iostream> using namespace std; template<typename T> void swap(T a, T b)
{ T temp = a;
 a = b;
 b = temp; }

  int main() {
  int x, y;
 cin >> x >> y;
  swap(x, y);
 cout << x << y;
  return 0; }
```

- ○ 9 14
- ◉ 914
- ○ 149
- ○ error

What feature of OOP is shown when classes have different functionality but share the same interfaces? * 1 point

- ○ Inheritance
- ◉ Polymorphism
- ○ Modularity
- ○ All of these