Q1. Will the following code compile? If it does not, state the errors. If it does compile, write the output.

University.h:

```cpp
#include <string>
namespace lums
{
    struct Student
    {
        string name;
        int id;
        Field *major;

        Student(string, int);
    };

    struct Field
    {
        string name;
        Field();
    };
}

struct Student
{
    string name;
    int id;
    Student(string, int);
};
```

University.cc:

```cpp
#include University.h;
#include <iostream>
using namespace std;
using namespace lums;

Student::Student(string name, int id)
{
    this.name = name;
    this.id = id;
    cout<"\nIn Student cstor"<endl;
    cout<"\n"<endl;
}

lums::Field::Field()
{
    name = "Computer science";
}

lums::Student::Student(string name, int id)
{
    this.name = name;
    this.id = id;
    cout<"\nMajor is "<major.name<endl;
    cout<"\nIn lums Student cstor"<endl;
}

int main()
{
    cout<"\nEnter name of student:"<endl;
    string name;
    cin.getline(name);
    cout<"\nEnter id"<endl;
    int id;
    cin>>id;
    lums::Student s1 = lums::Student(name, id);
    Student s = Student(name, id);
    return 0;
}
```

The code will not compile.

Errors:
1. Field struct must be defined above Student.
2. this is a pointer so must be used as this->
3. Compiler will not be able to distinguish between Student and Student in lums namespace due to the use in .cc file
4. major is un initialized pointer
5. cin.getline takes char * not string as argument.
Q2. Will the following code compile? If it does not, state the errors. If it does compile, write the output.

Functions.cc:
```cpp
#include <iostream>

using namespace std;

void printData(long i)
{
    cout<<"In long printData "<<i<<endl;
}

void printData(int i)
{
    cout<<"In int printData 
"<<i<<endl;
}

void printData(float i)
{
    cout<<"In float printData 
"<<i<<endl;
}

int main()
{
    int i = 10;

    printData(i);
    printData((float)i);
    printData(20L);
    return 0;
}
```

The code will compile. Output is:

In int printData 10
In float printData 10
In long PrintData 20
Q3. Redo Q2 with one more function in Functions.cc:

```cpp
bool printData(int i)
{
    cout<<”in bool printData ”<<i<<endl;
    return true;
}
```

The code will not compile as the compiler will not be able to differentiate between the functions `void printData(int i)` and `bool printData(int i)` as both take the same number and type of arguments.

Q4. We know that compiler provides us with a default constructor. State as many reasons as possible due to which you will need to write a constructor of your own.

- when you want to bring the initialization closer to declaration
- when you have references as member variables in the class definition
- when you have const member variables in class definition
- when your class contains objects of classes that do not have a default constructor