Exercises – Inheritance

1. Create a program that represents different types of vehicles that we can use. Starting off with a base class called TransportVehicle, think of the variables and functions that every vehicle has. These are the properties that will be common to all transport vehicles.

   Now that the base class has been written, derive more specific classes from a basic transport vehicle. These could be things like Car, Plane or Bicycle. Each derived class should have data and functionality unique to the derived class. For example, a car can have a measurement of how powerful it is, while a bicycle can have a measure of how many gears it has. The functionality doesn't have to be completely fleshed out, but should provide an idea of what it will do in the future when it does get filled out.

   Remember to place class definitions in separate .h files and function implementations in separate .cpp files. Make sure to use the correct access specifiers to provide the right level of access to variables and functions.

2. What will the following program display?

```
#include <iostream>

class Sky
{
private:
	int m_Number;

public:
	Sky()
	{
		m_Number = 0;
		printf("ENTERING THE SKY\n");
	}

	Sky(int a_Number)
	{
		m_Number = a_Number;
		printf("ENTERING THE SKY %i TIMES\n",
m_Number);
	}

	~Sky()
	{
		printf("LEAVING THE SKY\n");
	}
};

class Ground : public Sky
{
public:
	Ground(int a_Number)
		: Sky(a_Number)
	{
		printf("ENTERING THE GROUND %i TIMES\n",
a_Number);
	}

	Ground()
	{
		printf("ENTERING THE GROUND\n");
	}

	~Ground()
	{
		printf("ENTERING THE GROUND\n");
	}
};

void main()
{
	Ground groundObject;
	system("pause");
}
```

3. A retail store has a preferred customer plan where customers may earn discounts on all their purchases. The amount of a customer's discount is determined by the amount of the customer's cumulative purchases in the store. When a customer spends over:
    a. $500, they get a 5% discount on all future purchases
    b. $1000, they get a 6% discount on all future purchases
    c. $1500, they get a 7% discount on all future purchases
    d. $2000, they get a 10% discount on all future purchases

   Design a class named PreferredCustomer, which is derived from the CustomerData class. The PreferredCustomer class should have the following member variables:
    - purchasesAmount
    - discountLevel

   The purchasesAmount variable holds the accumulative cost of purchases the customer has made to date. The discountLevel variable holds the correct discount percentage according to the scheme described above.