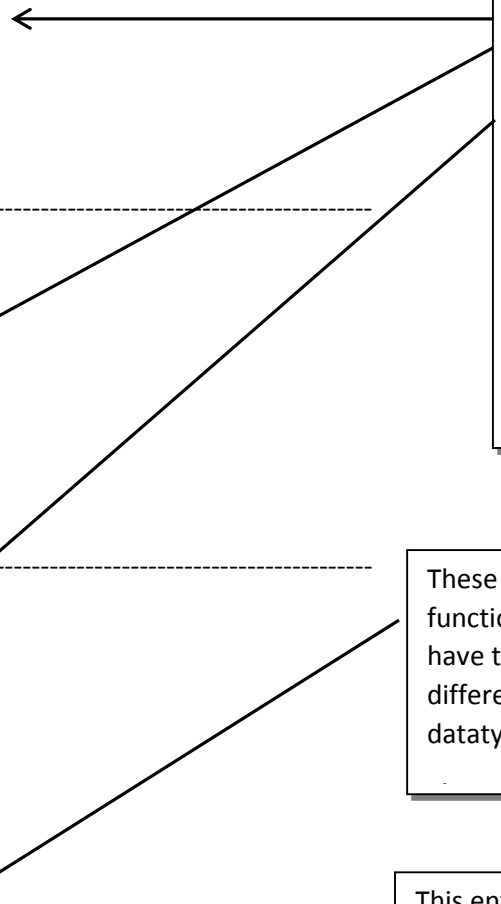
```
using namespace std;
#include <iostream>
#include <string>

//--------------------------------------------------------------------------------
class rockrecord
{
public:
        void getinfo();
        string name;
        int power;
        bool altweapon;
};

//--------------------------------------------------------------------------------
class paperrecord
{
public:
        void getinfo();
        string name;
        int power;
        bool altweapon;
};

//--------------------------------------------------------------------------------
class scissorsrecord
{
public:
        void getinfo();
        string name;
        int power;
        bool altweapon;
};

void match(rockrecord rockofages, paperrecord toil);
void match(rockrecord rockofages, scissorsrecord edcutup);
void match(scissorsrecord edcutup, paperrecord toil);

//--------------------------------------------------------------------------------
```



These are the layouts for the prototypes for the classes of rock paper scissors.

These are user defined data types.

The functions getinfo() are functions which are part of each of the classes, but they are three different functions as far as the compiler is concerned.

These are ***overloaded*** function names since they have the same name but for different user defined datatypes.

This entire page is just the prototytpes for the rest of the program.

```
int main()
{       int i;
        rockrecord rockofages;
        paperrecord toil;
        scissorsrecord edcutup;


        edcutup.getinfo();
        rockofages.getinfo();
        toil.getinfo();

        match(rockofages, edcutup);
        match(rockofages, toil);
        match(edcutup, toil);

        cin >> i;
        return (0);
}
```

This the main program.

These statements actually declare variables which of the datatype previously defined by the user.

These actually call (activate or invoke are other terms) the functions inside each datatype previously defined by the user.

These actually call (activate or invoke are other terms) the ***overloaded*** functions.

```
//---------------------------------------------------------------------------

void rockrecord::getinfo()
{
        char userchoice;
        cout << " Enter name for rock ";
        cin >> name;
        cout << " Enter power  ==> ";
        cin >> power;
        cout << " Alt weapon needed ==> ";
        cin >> userchoice;
        if (userchoice == 'Y')
                altweapon = true;
        else altweapon = false;
}

//---------------------------------------------------------------------------

void paperrecord::getinfo()
{
        char userchoice;
        cout << " Enter name for paper ";
        cin >> name;
        cout << " Enter power  ==> ";
        cin >> power;
        cout << " Alt weapon needed ==> ";
        cin >> userchoice;
        if (userchoice == 'Y')
                altweapon = true;
        else altweapon = false;
}


//---------------------------------------------------------------------------

void scissorsrecord::getinfo()
{
        char userchoice;
        cout << " Enter name scissors ";
        cin >> name;
        cout << " Enter power  ==> ";
        cin >> power;
        cout << " Alt weapon needed ==> ";
        cin >> userchoice;
        if (userchoice == 'Y')
                altweapon = true;
        else altweapon = false;
}
```

These are the function definitions which has the actual code which executes if the functions are called or invoked.

The altweapon for each class might be:

Rockrecord has a paper shredder for dealing with paper.

Scissorsrecord might have a jackhammer for dealing with rocks.

Paperrecord might have a special metal cover or a rock to break scissors.

```
//-------------------------------------------------------------------------------

void match(rockrecord rockofages, paperrecord toil)
{
        cout << "\nRock loses to paper\n\n";
}

//-------------------------------------------------------------------------------

void match(rockrecord rockofages, scissorsrecord edcutup)
{
        cout << "\nRock breaks scissors\n\n";
}

//-------------------------------------------------------------------------------

void match(scissorsrecord edcutup, paperrecord toil)
{
        cout << "\nScissors cuts paper\n\n";
}
```

These are the definitions of the ***overloaded*** functions.

They are overloaded since the datatypes of each function are different; but the name of the function is the same.

As far as the compiler is concerned, these are three different functions.