W2P1:

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

void printStarPattern(int lines) {
	int i=lines;
	while(i>0) {
		int j=0;
		while(j<i) {
			cout<<"* ";
			j++;
		}
		cout<<endl;
		i--;
	}
}

void printAlphabetPattern(int lines) {
	int i=lines;
	while(i>0) {
		char ch='A';
		int j=0;
		while(j<i) {
			cout<<ch<<" ";
			ch++;
			j++;
		}
		cout<<endl;
		i--;
	}
}
```

```
void printFloydsTriangle(int lines) {
	int num=1;
	int i=1;
	while(i<=lines) {
		int j=0;
		while(j<i) {
			cout<<num<<" ";
			num++;
			j++;
		}
		}
		cout<<endl;
		i++;
	}
}

int main() {
	int lines,choice;
	cout<<"Enter the number of lines:";
	cin>>lines;
	cout<<"Choose the pattern to print (1 for Stars, 2 for Alphabets, 3 for Floyd's Triangle):";
	cin>>choice;
	switch(choice) {
		case 1:
			printStarPattern(lines);
			break;
		case 2:
			printAlphabetPattern(lines);
			break;
		case 3:
```

```
            printFloydsTriangle(lines);
            break;
        default:
            cout<<"Invalid choice!"<<endl;
            break;
    }
    return 0;
}
```

W2P2

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

bool isPerfectCube(int n) {
    if(n<0)return false;
    int guess=0;
    while(guess*guess*guess<=n) {
        if(guess*guess*guess==n)return true;
        guess++;
    }
    return false;
}

int main() {
    int number;
    cout<<"Enter a number:";
    cin>>number;
    if(isPerfectCube(number))cout<<number<<" is a perfect cube."<<endl;
    else cout<<number<<" is not a perfect cube."<<endl;
    return 0;}
```

W2P3

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

void decimalToBinary(int n) {
	if(n==0) {
		cout<<"0";
		return;
	}
	string binary="";
	while(n>0) {
		binary=to_string(n%2)+binary;
		n/=2;
	}
	cout<<binary;
}

int main() {
	int number;
	cout<<"Enter a decimal number:";
	cin>>number;
	cout<<"Binary equivalent:";
	decimalToBinary(number);
	cout<<endl;
	return 0;
}
```

W2P4

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

```
void generateFibonacci(int n) {
    int a=0,b=1;
    for(int i=0;i<n;i++) {
        cout<<a<<" ";
        int next=a+b;
        a=b;
        b=next;
    }
}

int main() {
    int n;
    cout<<"Enter the number of terms:";
    cin>>n;
    cout<<"Fibonacci sequence:"<<endl;
    generateFibonacci(n);
    cout<<endl;
    return 0;
}
```

W2P5

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

bool isPrime(int num) {
    if(num<=1)return false;
    for(int i=2;i*i<=num;i++) {
        if(num%i==0)return false;
    }
    return true;
```

```
}

int main() {
	int lower, upper;
	cout<<"Enter the lower limit:";
	cin>>lower;
	cout<<"Enter the upper limit:";
	cin>>upper;
	cout<<"Prime numbers between "<<lower<<" and "<<upper<<":"<<endl;
	for(int i=lower;i<=upper;i++) {
		if(isPrime(i))cout<<i<<" ";
	}
	cout<<endl;
	return 0;
}
```

W2P6

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

void makeChange(int amount) {
	int coins[]={500,200,100,50,20,10,5,1};
	int numCoins=sizeof(coins)/sizeof(coins[0]);
	cout<<"Change for "<<amount<<":"<<endl;
	for(int i=0;i<numCoins;i++) {
		int count=amount/coins[i];
		if(count>0) {
			cout<<coins[i]<<": "<<count<<endl;
			amount-=count*coins[i];
		}
	}
```

```
}

int main() {
    int bill,cash;
    cout<<"Enter the total bill amount:";
    cin>>bill;
    cout<<"Enter the cash paid:";
    cin>>cash;
    int balance=cash-bill;
    if(balance<0) {
        cout<<"Insufficient cash paid."<<endl;
        return 1;
    }
    makeChange(balance);
    return 0;
}
```

W2P7

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

struct BankAccount {
    string customerName;
    string accountNumber;
    string accountType;
    double balance;

    void deposit(double amount) {
        if(amount>0) {
            balance+=amount;
```

```
            cout<<"Deposited ?"<<amount<<". New balance is ?"<<balance<<"."<<endl;
        } else {
            cout<<"Invalid deposit amount."<<endl;
        }
    }

    void withdraw(double amount) {
        if(amount>0) {
            if(amount<=balance) {
                balance-=amount;
                cout<<"Withdrew ?"<<amount<<". New balance is ?"<<balance<<"."<<endl;
            } else {
                cout<<"Insufficient balance."<<endl;
            }
        } else {
            cout<<"Invalid withdrawal amount."<<endl;
        }
    }
};

int main() {
    BankAccount account;
    cout<<"Enter customer name: ";
    getline(cin,account.customerName);
    cout<<"Enter account number: ";
    getline(cin,account.accountNumber);
    cout<<"Enter account type (Savings/Fixed/Current): ";
getline(cin,account.accountType);
    cout<<"Enter initial balance: ?";
    cin>>account.balance;
```

```
int choice;
double amount;

do {
    cout<<"\nBank Account Operations Menu:"<<endl;
    cout<<"1. Deposit"<<endl;
    cout<<"2. Withdraw"<<endl;
    cout<<"3. Exit"<<endl;
    cout<<"Enter your choice: ";
    cin>>choice;

    switch(choice) {
        case 1:
            cout<<"Enter amount to deposit: ?";
            cin>>amount;
            account.deposit(amount);
            break;
        case 2:
            cout<<"Enter amount to withdraw: ?";
            cin>>amount;
            account.withdraw(amount);
            break;
        case 3:
            cout<<"Exiting."<<endl;
            break;
        default:
            cout<<"Invalid choice."<<endl;
            break;
    }
} while(choice!=3);
```

```
    return 0;
}
```

Week2Bonus

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

int countOnesInBinary(int num) {
    int count=0;
    while(num) {
        count+=num&1;
        num>>=1;
    }
    return count;
}

int main() {
    int n;
    cout<<"Enter the number of terms to display:";
    cin>>n;
    int found=0;
    int num=0;
    while(found<n) {
        if(countOnesInBinary(num)%2==0) {
            cout<<num<<" ";
            found++;
        }
        num++;
    }
    cout<<endl;
    return 0;
```

}