typedef struct ClassSeat\_struct {

char firstName[50];

char lastName[50];

} Class;

void SeatMakeEmpty(Class \*math) {

strcpy((\*math).firstName, "unfilled");

strcpy((\*math).lastName, "unfilled");

}

void EngSeatMakeEmpty(Class \*eng) {

strcpy((\*eng).firstName, "unfilled");

strcpy((\*eng).lastName, "unfilled");

}

void SciSeatMakeEmpty(Class \*sci) {

strcpy((\*sci).firstName, "unfilled");

strcpy((\*sci).lastName, "unfilled");

}

void HistSeatMakeEmpty(Class \*hist) {

strcpy((\*hist).firstName, "unfilled");

strcpy((\*hist).lastName, "unfilled");

}

bool SeatIsEmpty(Class math) {

return (strcmp(math.firstName, "unfilled") == 0);

}

bool EngSeatIsEmpty(Class eng) {

return (strcmp(eng.firstName, "unfilled") == 0);

}

bool SciSeatIsEmpty(Class sci) {

return (strcmp(sci.firstName, "unfilled") == 0);

}

bool HistSeatIsEmpty(Class hist) {

return (strcmp(hist.firstName, "unfilled") == 0);

}

void SeatPrint(Class math) {

printf("%s ", math.firstName);

printf("%s ", math.lastName);

printf("\n");

}

void EngSeatPrint(Class eng) {

printf("%s ", eng.firstName);

printf("%s ", eng.lastName);

printf("\n");

}

void SciSeatPrint(Class sci) {

printf("%s ", sci.firstName);

printf("%s ", sci.lastName);

printf("\n");

}

void HistSeatPrint(Class hist) {

printf("%s ", hist.firstName);

printf("%s ", hist.lastName);

printf("\n");

}

void SeatsMakeEmpty(Class math[], int numSeats) {

int i;

for (i = 0; i < numSeats; ++i) {

SeatMakeEmpty(&math[i]);

}

}

void EngSeatsMakeEmpty(Class eng[], int numSeats) {

int i;

for (i = 0; i < numSeats; ++i) {

EngSeatMakeEmpty(&eng[i]);

}

}

void SciSeatsMakeEmpty(Class sci[], int numSeats) {

int i;

for (i = 0; i < numSeats; ++i) {

SciSeatMakeEmpty(&sci[i]);

}

}

void HistSeatsMakeEmpty(Class hist[], int numSeats) {

int i;

for (i = 0; i < numSeats; ++i) {

HistSeatMakeEmpty(&hist[i]);

}

}

void SeatsPrint(Class math[], int numSeats) {

int i;

for (i = 0; i < numSeats; ++i) {

printf("%d: ", i);

SeatPrint(math[i]);

}

}

void EngSeatsPrint(Class eng[], int numSeats) {

int i;

for (i = 0; i < numSeats; ++i) {

printf("%d: ", i);

SeatPrint(eng[i]);

}

}

void SciSeatsPrint(Class sci[], int numSeats) {

int i;

for (i = 0; i < numSeats; ++i) {

printf("%d: ", i);

SeatPrint(sci[i]);

}

}

void HistSeatsPrint(Class hist[], int numSeats) {

int i;

for (i = 0; i < numSeats; ++i) {

printf("%d: ", i);

SeatPrint(hist[i]);

}

}

int main(void) {

const int NUM\_SEATS = 7;

char userKey;

int seatNum;

Class math[NUM\_SEATS];

Class eng[NUM\_SEATS];

Class sci[NUM\_SEATS];

Class hist[NUM\_SEATS];

Class currSeat;

SeatsMakeEmpty(math, NUM\_SEATS);

SeatsMakeEmpty(eng, NUM\_SEATS);

SeatsMakeEmpty(sci, NUM\_SEATS);

SeatsMakeEmpty(hist, NUM\_SEATS);

int i; // determine what seat number

int j; // determine what class

while (userKey != 'q') {

printf("Print classes = p\nInput registration = r\nExit registration = q\n Enter command (p/r/q): ");

scanf(" %c", &userKey);

if (userKey == 'p') { // Print classes

printf("List of Classes\n 0 = Calculus II\n 1 = Programming for Engineers\n 2 = Physics II\n 3 = SMC Nature\nSelect which class to look at: ");

scanf("%d", &j);

switch(j){

case 0:

printf(" Calculus II \n");

SeatsPrint(math, NUM\_SEATS);

printf("\n");

break;

case 1:

printf(" Programming for Engineers \n");

SeatsPrint(eng, NUM\_SEATS);

printf("\n");

break;

case 2:

printf(" Physics II \n");

SeatsPrint(sci, NUM\_SEATS);

printf("\n");

break;

case 3:

printf(" SMC Nature \n");

SeatsPrint(hist, NUM\_SEATS);

printf("\n");

break;

}

}

else if (userKey == 'r') {

printf("\n0 = Calculus II\n1 = Programming for Engineers\n2 = Physics II\n3 = SMC Nature\n Pick what course to register for: ");

scanf("%d", &i);

switch(i){

case 0: // Calculus 2

printf("Register for Calculus II:\n");

printf("Enter seat num: ");

scanf("%d", &seatNum); // user inputs a seat number

if (!SeatIsEmpty(math[seatNum])) {

printf("Seat not unfilled.\n\n");

break;

}

else {

printf("Enter first name: ");

scanf("%s", currSeat.firstName);

printf("Enter last name: ");

scanf("%s", currSeat.lastName);

math[seatNum] = currSeat;

printf("Completed.\n\n");

break;

}

case 1: // Eng Prog

printf("Register for Programming for Engineers:\n");

printf("Enter seat num: ");

scanf("%d", &seatNum);

if (!SeatIsEmpty(eng[seatNum])) {

printf("Seat not unfilled.\n\n");

break;

}

else {

printf("Enter first name: ");

scanf("%s", currSeat.firstName);

printf("Enter last name: ");

scanf("%s", currSeat.lastName);

eng[seatNum] = currSeat;

printf("Completed.\n\n");

break;

}

case 2: // Physics 2

printf("Register for Physics II:\n");

printf("Enter seat num: ");

scanf("%d", &seatNum);

if (!SeatIsEmpty(sci[seatNum])) {

printf("Seat not unfilled.\n\n");

break;

}

else {

printf("Enter first name: ");

scanf("%s", currSeat.firstName);

printf("Enter last name: ");

scanf("%s", currSeat.lastName);

sci[seatNum] = currSeat;

printf("Completed.\n\n");

break;

}

case 3: // SMC Nature

printf("Register for SMC Nature:\n");

printf("Enter seat num: ");

scanf("%d", &seatNum);

if (!SeatIsEmpty(hist[seatNum])) {

printf("Seat not unfilled.\n\n");

break;

}

else {

printf("Enter first name: ");

scanf("%s", currSeat.firstName);

printf("Enter last name: ");

scanf("%s", currSeat.lastName);

hist[seatNum] = currSeat;

printf("Completed.\n\n");

break;

}

}

}

else if (userKey == 'q') {

printf("Exiting registration.\n");

}

else {

printf("No, you cannot do that.\n\n");

}

}

return 0;

}